

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ О.В. Коваль
(підпис) (ініціали, прізвище)
“ ” _____ 2019р.

Магістерська дисертація

зі спеціальності - 122 Комп'ютерні науки
за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання
процесів і систем
а тему: Розробка системи супроводження та аналізу навчального процесу
пеціалістів науково-педагогічних кадрів вищої кваліфікації

Виконав (-ла): студент (-ка) 6 курсу, групи ТМ-81мп

Гумінський Артем Миколайович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник к.т.н., доцент Коваль О.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ - 2019

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 122 Комп'ютерні науки

за спеціалізацією – Комп'ютерний моніторинг та геометричне моделювання процесів і систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

О.В. Коваль

(підпис)

” ” _____ 2019р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Гумінському Артему Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації: «Система супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації»

Науковий керівник Коваль Олександр Васильович, к.т.н., доцент

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджені наказом по університету від ” ” __ 201__ р. № __

2. Строк подання студентом дисертації 9 грудня 2019р.

3. Вихідні дані до роботи магістерської дисертації

мова програмування TypeScript та Kotlin, середовище розробки Microsoft Visual Code Studio та Android Studio, фреймворк Angular, фреймворк Android

4. Перелік завдань, які потрібно розробити

проаналізувати проблему навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації, розробити систему супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації

5. Перелік ілюстративного матеріалу

схеми архітектури додатку, знімки інтерфейсу додатку

6. Орієнтований перелік публікацій

Міжнародна наукова інтернет-конференція на тему «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення» (випуск 44). Система

супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації

7. Дата видачі завдання ”10” грудня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітки
1.	Затвердження теми роботи	10.12.18	
2.	Вивчення та аналіз задачі	11.02 – 11.03.19	
3.	Розробка архітектури та загальної структури системи	12.03 – 31.05.19	
4.	Розробка структур окремих підсистем	1.06 – 31.06.19	
5.	Програмна реалізація системи	03.09 – 22.10.19	
6.	Оформлення пояснювальної записки	01.11 – 07.12.19	
7.	Захист програмного продукту	23.10.19	
8.	Передзахист	20.11.19	
9.	Захист	16.12.19	

Студент _____
(підпис)

Науковий керівник _____
(підпис)

Гумінський А. М.
(прізвище та ініціали)

Коваль О. В.
(прізвище та ініціали)

РЕФЕРАТ

Магістерська дисертація складається зі вступу, п'яти розділів, висновку, переліку посилань з 21 найменування, одного додатку, і містить 32 рисунків, 21 таблиць. Повний обсяг магістерської дисертації складає 92 сторінки, з яких перелік посилань займає 2 сторінки, додаток – 6 сторінок.

Актуальність теми. Існуючі на ринку програмні продукти, як правило, слабо враховують яскраво виражену галузеву специфіку ринку освіти. До того ж, через брак ресурсів проекти комплексної автоматизації в ВНЗ часто гальмуються і не доводяться до кінця. Деякі ВНЗ йдуть шляхом створення власних розробок силами фахівців ІТ-підрозділів, що дозволяє створювати рішення, що враховують особливості бізнес-процесів конкретного закладу. Але плюси такого підходу дуже часто повністю нівелюються великою залежністю від команди програмістів. ВНЗ сьогодні працюють в умовах жорсткої конкуренції, більш того - змушені щодня підтверджувати свою затребуваність на ринку освітніх послуг і фінансову спроможність. Отже, вони повинні підходити до питань управління як повноцінні бізнес-одиниці, в тому числі - в плані управлінського інструментарію.

Метою дослідження та розробки є створення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації.

Об'єктом дослідження є комп'ютерні інформаційні системи і технології.

Предметом дослідження є комп'ютерні інформаційні системи і технології, а саме - веб-розробка та мобільна розробка.

Для досягнення мети в рамках дослідження необхідно виконати наступні завдання:

- сформулювати науково-технічну проблему, визначити об'єкт, предмет та мету дослідження, проаналізувати стан рішення проблеми за матеріалами вітчизняних і зарубіжних публікацій, обґрунтувати цілі дослідження

- проаналізувати можливі методи досліджень і варіантів рішення завдання, обґрунтувати вибір технічного рішення;
- провести науковий аналіз і узагальнення фактичного матеріалу, який використовується в процесі дослідження або виконання розрахунків щодо обраного технічного рішення;
- викласти отримані результати та оцінити їхнє прикладне значення.

Наукова новизна отриманих результатів. Найбільш суттєвими науковими результатами виконаної магістерської дисертації є:

- удосконалено систему навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації за рахунок створення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації;
- набуто подальшого розвитку використання автоматизованих систем у освітньому процесі.

Практичне значення отриманих результатів: засвоєння навичок правильної постановки проблеми та обґрунтування її актуальності, формулювання мети і завдань дослідження, побудови логічного плану й оптимальної структури магістерської роботи, робота з літературними джерелами та статистичною інформацією, аналізу та оцінки різних аспектів обраного проекту, обґрунтування власних узагальнень, висновків і пропозицій; використання набутих у процесі навчання теоретичних знань, практичних навичок для створення повноцінного програмного продукту.

Апробація результатів дисертації. Результати досліджень оприлюднені на: Міжнародній науковій інтернет-конференції на тему: «Інформаційне суспільство: технологічні, економічні та технічні аспекти встановлення» (випуск 44) [Додаток А].

Ключові слова: НАВЧАННЯ, АСПІРАНТ, ВИКЛАДАЧ, АДМІНІСТРАЦІЯ, КРОСПЛАТФОРМНІСТЬ, МОБІЛЬНИЙ ДОДАТОК, ANDROID ДОДАТОК, ВЕБ-ДОДАТОК, ХМАРНА ТЕХНОЛОГІЯ.

ABSTRACT

Master's Thesis consists of an introduction, five sections, a conclusion, a list of references of 21 titles, one application, and contains 32 figures, 21 tables. The full volume of the master's thesis is 92 pages, of which the list of links occupies 2 pages, the application - 6 pages.

Actuality of theme. Existing software products, as a rule, have little regard for the pronounced industry specificity of the education market. In addition, due to lack of resources, complex automation projects in universities are often slowed down and not completed. Some universities go through the creation of their own developments by specialists of IT-departments, which allows to create solutions that take into account the peculiarities of business processes of a particular institution. But the benefits of this approach are very often completely offset by a great dependence on the programming team. Universities today operate in conditions of fierce competition, moreover - they are forced to confirm their demand on the market of educational services and financial capacity on a daily basis. Therefore, they should approach management issues as a full-fledged business unit, including in terms of management tools.

The purpose of the study and development is to create a system of support and analysis of the educational process of specialists of scientific and pedagogical staff of higher qualification.

The object of study is computer information systems and technologies.

The subject of the study is computer information systems and technologies, namely web development and mobile development.

In order to achieve the goal within the study, the following **tasks must be performed:**

- formulate a scientific and technical problem, determine the object, object and purpose of the research, analyze the state of the solution of the problem by the materials of domestic and foreign publications, substantiate the goals of the research;

- analyze possible research methods and options for solving the problem, justify the choice of technical solution;
- to conduct scientific analysis and generalization of the actual material used in the process of research or calculation of the selected technical solution;
- state the results obtained and evaluate their applied value.

Scientific novelty of the obtained results. The most significant scientific results of the completed master's thesis are:

- the system of educational process of specialists of scientific and pedagogical staff of higher qualification was improved by creating a system of support and analysis of the educational process of specialists of scientific and pedagogical staff of higher qualification;
- the use of automated systems in the educational process was further developed.

The practical significance of the obtained results: mastering the skills of correct statement of the problem and substantiation of its relevance, formulation of the purpose and tasks of the research, construction of a logical plan and optimal structure of the master's work, work with literary sources and statistical information, analysis and evaluation of various aspects of the selected project, substantiation of their own generalizations conclusions and suggestions; use of theoretical knowledge and practical skills acquired in the process of training to create a complete software product.

Testing the results of the thesis. The results of the research were published at: International Scientific Internet Conference on "Information Society: Technological, Economic and Technical Aspects of Installation" (Issue 44) [Appendix A].

Keywords: STUDYING, POSTGRADUATE, TEACHER, ADMINISTRATION, CROSS-PLATFORM, MOBILE APP, ANDROID APP, WEB APP, CLOUD TECHNOLOGY.

ЗМІСТ

Перелік умовних позначень і скорочень.....	10
Вступ.....	11
1. Задача створення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації	13
1.1. Загальний аналіз проблеми розробки системи.....	15
1.2. Призначення розробки.....	16
1.3. Висновки до розділу 1	18
2. Огляд існуючих програмних рішень	19
2.1. АСУ «ВНЗ».....	19
2.2. ПП «Політек - СОФТ»	21
2.3 Програмний комплекс «АЛЬМА - МАТЕР».....	23
2.4. Висновки до розділу 2	25
3. Методи програмної реалізації системи	26
3.1. Середовище розробки	27
3.2. Мова програмування Kotlin	28
3.3. Бібліотека ін'єкції залежностей Dagger	30
3.3.1. Основна робота принципу ін'єкції залежностей	30
3.3.2. Використання Dagger	31
3.4. Бібліотека ReactiveX	32
3.4.1. Гнучкі Observables.....	33
3.4.2. Використання бібліотек RxJava та RxJs.....	34
3.4.3. Використання бібліотеки Retrofit	36
3.5. Опис використання платформи Firestore	37
3.6. Висновки до розділу 3	38
4. Опис функціоналу системи за основними типами користувачів	39

4.1. Опис функціоналу системи з точки зору адміністратора	40
4.2. Опис функціоналу системи з точки зору викладача.....	41
4.3. Опис функціоналу системи з точки зору користувача (аспіранта).....	42
4.4. Приклад взаємодії з системою	43
4.5. Висновки до розділу 4	44
5. Методика роботи користувача з системою	45
5.1. Опис спільних частин при взаємодії з програмним компонентом	45
5.2. Опис взаємодії з модулем студента.....	46
5.3. Опис взаємодії з модулем викладача	52
5.4. Опис взаємодії з модулем адміністратора	54
5.5. Висновки до розділу 5	59
6. Розроблення стартап-проекту	60
6.1. Опис ідеї системи	60
6.2. Технологічний аудит проекту	67
6.3. Аналіз ринкових можливостей запуску стартап - проекту	69
6.4. Розроблення ринкової стратегії проекту	77
6.5. Розроблення маркетингової програми стартап проекту	80
6.6. Висновки до розділу 6	83
Висновки	85
Список використаних джерел	86
Додаток 1. Апробація.....	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

НТУУ "КПІ" - Національний технічний університет України "Київський політехнічний інститут"

ВНЗ - вищий навчальний заклад

БД - база даних

АСУ - автоматизована система управління

ІВС - інформаційно-виробнича система

ПП - приватне підприємство

ПЗ - програмне забезпечення

SDK (від англ. software Development Kit) - набір із засобів розробки, утиліт і документації, що дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи

JVM (від англ. Java Virtual Machine) - Віртуальна джава машина

APK (від англ. Android Package) - формат архівних файлів- додатків для Android

SVN (від англ. Subversion) - вільна система управління версіями

AVD (від англ. Android Virtual Device) - емулятор смартфона на базі операційної системи Android

ВСТУП

Загальна політика інформатизації є складовою частиною соціально-економічної діяльності університету і спрямовується на раціональне використання освітнього та науково-технічного потенціалу, матеріально-технічних і фінансових ресурсів для забезпечення необхідного рівня надання інформаційних послуг, відповідно до загальної стратегії розвитку університету.

Концепція інформатизації університету забезпечує задоволення інформаційних потреб користувачів, зацікавлених в діяльності університету, через надання інформаційних послуг. Зацікавленими користувачами є студенти, керівництво та працівники університету, фізичні особи, підприємства й організації, суспільство, міжнародні організації, тощо [1].

В рамках даної роботи пропонується висвітлення та аналіз проблеми потреб спеціалістів науково-педагогічних кадрів вищої кваліфікації, які досі залишаються незадоволеними.

На даний момент, користувачі цього напрямку мають досить обмежені можливості в рамках пошуку, перегляду, запиту, обміну, розміщення та аналізу необхідної інформації на базі існуючого веб-рішення [2], що надається університетом. Помітно, що даному ресурсу необхідні оновлення функціоналу, а також розробка зручного та сучасного рішення у вигляді мобільного додатку на базі технології Android.

В університеті НТУУ «КПІ імені Ігоря Сікорського» досі не існує комплексного рішення у вигляді системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації. Це є великою проблемою, адже спеціалісти даного напрямку не мають доступу до важливої та необхідної для навчання інформації, зібраної в одному місці зі зручним інтерфейсом.

Крім того, з іншого боку, адміністрація також не має змоги донести актуальну інформацію до студентів-спеціалістів, так як ще досі не було розроблено платформи

чи веб-сайту, для взаємодії як зі вступниками, так і з існуючими студентами даного напрямку. Це призводить до того, що обидві сторони втрачають дорогоцінний час на фізичне відвідування Відділу Аспірантури університету для отримання необхідних даних, які, на жаль, можуть бути неактуальними вже через деякий час.

Доцільність проведення дослідження полягає в удосконаленні зв'язків управління процесами проектування й організації діяльності університету, зокрема – Відділу Аспірантури.

1. ЗАДАЧА СТВОРЕННЯ СИСТЕМИ СУПРОВОДЖЕННЯ ТА АНАЛІЗУ НАВЧАЛЬНОГО ПРОЦЕСУ СПЕЦІАЛІСТІВ НАУКОВО-ПЕДАГОГІЧНИХ КАДРІВ ВИЩОЇ КВАЛІФІКАЦІЇ

Основним завданням системи вищої освіти є задоволення потреб ринку праці у фахівцях потрібного профілю. При цьому, фахівці, що випускаються, можуть мати різну якість підготовки, від якої залежить те, наскільки вони будуть відповідати встановленим вимогам на ринку праці. Розвиток суспільства, зокрема науки і техніки, постійно вносить корективи в зміст вищої освіти. Спеціаліст кожного нового випуску того чи іншого навчального закладу завжди повинен мати більш високий рівень підготовки, ніж фахівець попереднього випуску.

На ринку освіти сьогодні склалася парадоксальна ситуація: будучи осередком інтелектуальних ресурсів країни, вищі навчальні заклади залишаються найменш автоматизованої галуззю. Автоматизація систем управління в більшості вузів носить клаптевий характер і, в кращому випадку, обмежується встановленням бухгалтерських та складських облікових програм.

Проблематика. Підготовка фахівців, що відповідають сучасним запитам, тягне за собою безперервне вдосконалення навчальних планів і навчального процесу для того, щоб вони завжди відповідали вимогам, що поставлені часом. У цих умовах вищі навчальні заклади зіштовхуються з важкою проблемою налагодження ефективного управління процесом навчання, яке б враховувало всі нюанси сьогодення. При цьому без сучасних інформаційних технологій обійтися неможливо.

Метою дослідження та розробки є створення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації.

Об'єктом дослідження є комп'ютерні інформаційні системи і технології.

Предметом дослідження є комп'ютерні інформаційні системи і технології, а саме - веб-розробка та мобільна розробка.

Цілями створення системи є:

1. Підвищення ефективності роботи Аспірантського структурного підрозділу університету та системи управління ВНЗ в цілому.
2. Забезпечення можливості роботи з корпоративними адміністративними даними.
3. Створення єдиного інформаційного простору ВНЗ для Аспірантів, Викладачів та Адміністрації.
4. Впорядкування інформаційних потоків.
5. Автоматизоване формування інтегрованої інформації.
6. Інформаційна підтримка системи управління якістю підготовки фахівців.

Актуальність. Існуючі на ринку програмні продукти, як правило, слабо враховують яскраво виражену галузеву специфіку ринку освіти. До того ж через брак ресурсів проекти комплексної автоматизації в ВНЗ часто гальмуються і не доводяться до кінця. Деякі ВНЗ йдуть шляхом створення власних розробок силами фахівців ІТ-підрозділів, що дозволяє створювати рішення, що враховують особливості бізнес-процесів конкретного закладу. Але плюси такого підходу дуже часто повністю нівелюються великою залежністю від команди програмістів. ВНЗ сьогодні працюють в умовах жорсткої конкуренції, більш того - змушені щодня підтверджувати свою затребуваність на ринку освітніх послуг і фінансову спроможність. Отже, вони повинні підходити до питань управління як повноцінні бізнес-одиниці, в тому числі - в плані управлінського інструментарію.

Для досягнення мети в рамках дослідження необхідно виконати наступні **завдання**:

- сформулювати науково-технічну проблему, визначити об'єкт, предмет та мету дослідження, проаналізувати стан рішення проблеми за матеріалами вітчизняних і зарубіжних публікацій, обґрунтувати цілі дослідження;

- проаналізувати можливі методи досліджень і варіантів рішення завдання, обґрунтувати вибір технічного рішення;
- провести науковий аналіз і узагальнення фактичного матеріалу, який використовується в процесі дослідження або виконання розрахунків щодо обраного технічного рішення;
- викласти отримані результати та оцінити їхнє прикладне значення.

Методи дослідження: порівняння; узагальнення; формалізація; аналіз; синтез; індукція і дедукція; методи емпіричного і теоретичного рівнів.

Практичне значення отриманих результатів: засвоєння навичок правильної постановки проблеми та обґрунтування її актуальності, формулювання мети і завдань дослідження, побудови логічного плану й оптимальної структури магістерської роботи, робота з літературними джерелами та статистичною інформацією, аналізу та оцінки різних аспектів обраного проекту, обґрунтування власних узагальнень, висновків і пропозицій; використання набутих у процесі навчання теоретичних знань, практичних навичок для створення повноцінного програмного продукту.

1.1 Загальний аналіз проблеми розробки системи

Метою розробки є реалізація комплексної системи у вигляді веб-додатку та мобільного застосунку на базі Android, що надає можливість відстеження та супроводження навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації.

Програмний продукт повинен надавати наступні можливості:

- визначення певних ролей для користувачів, а саме: адміністратор, викладач та користувач (аспірант);
- перегляд власного профілю для ролі викладача;
- перегляд занять та груп, де викладає викладач (для ролі викладача);
- виставлення оцінок/пропусків занять студентам за певними датами занять, що навчаються у викладача (для ролі викладача);

- створення груп для ролі адміністратора;
- редагування груп для ролі адміністратора;
- перегляд та пошук серед існуючих груп для ролі адміністратора;
- додавання викладачів до системи та пошук серед вже існуючих для ролі адміністратора;
- додавання навчальних предметів у створенні групи та назначення викладача для кожного навчального предмету для ролі адміністратора;
- перегляд власного профілю для ролі користувача;
- перегляд індивідуального плану для ролі користувача;
- перегляд занять (оцінок та присутності) за певним навчальним предметом для ролі користувача;
- перегляд розкладу занять для ролі користувача;
- перегляд статистики за власною успішністю для ролі користувача;
- доступ до мобільного додатку на операційній системі Android, що матиме всі вищеписані функції для ролі користувача.

Система має надавати чітко зрозумілий інтерфейс для відстеження та супроводження навчального процесу для користувачів будь-якої ролі. Система має бути розподілена на підсистеми (частини), що будуть взаємодіяти між собою, окрім цього підсистема для ролі користувача має інтегрувати в собі зовнішню систему «Розклад КПІ».

1.2 Призначення розробки

Система створена для автоматизації управління навчальним процесом спеціалістів науково-педагогічних кадрів вищої кваліфікації, адже сьогодні, за стрімкого розвитку інформаційних технологій, потоки інформації, якими оперує людина, збільшуються щодня. Окрім того, існує проблема актуалізації даних, адміністрація відділу аспірантури також не завжди має змогу донести актуальну інформацію до студентів-спеціалістів, так як ще досі не було розроблено платформи

чи веб-сайту, для взаємодії як зі вступниками, так і з існуючими студентами даного напрямку.

Саме тому у відділі аспірантури навчального закладу НТУУ «КПІ ім. Ігоря Сікорського" виникла потреба у керуванні даними, їх автоматизації та зручному представлені.

Слід зазначити, що розроблене рішення додатково може виконуватись на операційній системі Android з точки зору користувача(аспіранта).

Всю розробку системи можна поділити на декілька етапів:

1. Розробка адміністративної частини. Адміністративна частина передбачає сторінку авторизації адміністратора у системі, панель керування групами, з можливістю створення та редагування груп, пошуку по групах, та панель керування інформацією про викладачів, додавання до системи нових викладачів, призначення їм занять, пошук по викладачах. Для цієї частини має бути створено веб застосунок.
2. Розробка частини викладача. Частина викладача передбачає сторінку авторизації, сторінку перегляду власного профілю, сторінку відображення власних занять викладача. На сторінці занять мають відображатися картки з групами, де викладає викладач, картка групи повинна розгортатися та показувати список учнів в групі. Викладач повинен мати змогу додати певну відмітку для кожного учня в групі.
3. Розробка частини користувача. Частина користувача передбачає програмну реалізацію на двох застосунків, веб застосунку та Android застосунку. Кожен з них повинен мати сторінку авторизації, сторінку перегляду власного профілю, сторінку перегляду індивідуального плану з кількостями кредитів та атестаційною формою кожного з предметів, сторінку занять, з можливістю перегляду власних навчальних відміток, сторінку розкладу занять та сторінку власної статистики.

1.3 Висновки до Розділу 1

У цьому розділі визначено мету, об'єкт дослідження, предмет дослідження та завдання для вирішення проблеми супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації. У зв'язку проблемою налагодження ефективного управління процесом навчання було вирішено розробити три функціональні підсистеми:

- підсистема адміністратора;
- підсистема аспіранта;
- підсистема викладача.

У системі користувач переважно займає позицію спостерігача та має можливість лише слідкувати за перебігом події, у той час коли адміністратор системи та викладачі мають право на внесення змін до інформаційної бази шляхом взаємодії з веб-додатком.

2. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

На даний час в межах кафедри та університету не виявлено аналогічних програмних засобів, які забезпечують роботу супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації. Але якщо звернутись до мережі Інтернет, то можна знайти декілька веб-сторінок або програм, що надають схожий функціонал. Нижче розглянуто деякі можливості систем управління навчальним процесом: пакет програмних комплексів «Політек-софт» («Деканат» та «Персонал»), АСУ «ВНЗ», «ВНЗ АЛЬМА-МАТЕР».

Одразу слід наголосити на недоліках цих систем, а саме:

- складність супроводу системи;
- складний інтерфейс користувача;
- не підтримують універсальні апаратні платформи.

У нашому випадку є необхідним створення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації, як окремої комплексної інформаційної технології, задля автоматизації процесів діяльності ВНЗ, з урахуванням сучасної кредитно-модульної системи. Система повинна базуватися на основі клієнт-серверних середовищ.

Отже, головна мета даного розділу - дослідити та проаналізувати існуючі рішення автоматизованих систем управління навчальним процесом.

2.1 АСУ «ВНЗ»

Система АСУ «ВНЗ» слугує для автоматизованого керування вищим навчальним закладом, вона складається з двох підсистем, а саме: «Приймальна

комісія» та «Деканат». Кожна підсистема має окреме цільове призначення, розглянемо кожну з підсистем більш детально.

Підсистема «Деканат» надає можливість:

- формування підсумків заліково-екзаменаційної документації на період сесії;
- роздрукування і створення проектів наказів;
- створення заявок на виготовлення освітніх документів.

Підсистема «Приймальна комісія» надає можливість:

- оформлювати результати іспитів, що вносяться до системи адміністратором;
- перевіряти правильність введення інформації атестату;
- полегшити введення анкетних документів для абітурієнта.
- аналізувати результати вступних іспитів та будувати модель середнього та прохідного балу.

Система реалізована з використанням бази даних SQL та у середовищі Delphi (рис.1.1).

Система активно впроваджувалась у закладах освіти до 2006 року. На даний момент АСУ «ВНЗ» є частиною системи загальнодержавного рівня ІВС «ОСВІТА». Сьогодні ж більшість навчальних закладів використовують лише спеціалізований пакет DP 2000, що є безкоштовним та надає можливості для замовлення студентських квитків та формування освітніх документів. Розробники наголошують, що при використанні підсистеми «Деканат» адміністратори системи матимуть можливість здійснювати замовлення дипломів лише за декілька хвилин. Але автоматизація лише одного навчального процесу передбачає не лише формування освітніх документів, їх замовлення. Ще одним великим недоліком системи є відсутність підсистеми автоматизації навчального процесу. При купівлі вищим навчальним закладом системи АСУ «ВНЗ» - подальше оновлення підсистем не передбачено, перехід на нову систему здійснюється шляхом купівлі нової версії та імпортом/експортом даних.

Супроводження та налаштування програмного забезпечення здійснюється власними силами робітниками ВНЗ.

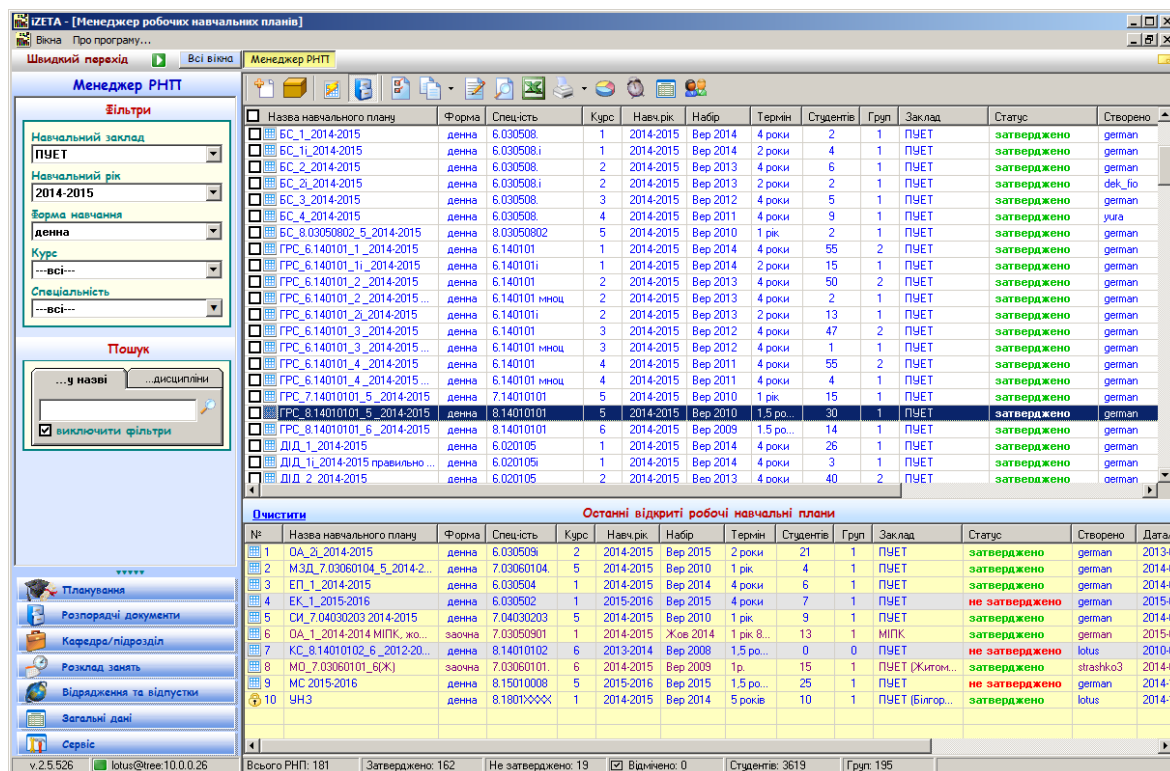


Рисунок 1.1 - Інтерфейс АСУ «ВНЗ»

2.2. ПП «Політек – СОФТ»

ПП «Політек – СОФТ» співпрацює з вищими навчальними закладами з 2000 року, про що свідчать ліцензії та сертифікати. ПЗ «Політек - СОФТ» охоплює більшу частину діяльності навчального закладу [2].

Пакети підсистем «Політек - СОФТ» надають наступні можливості:

- обмін даними між пакетами підсистеми;
- робота в локальній мережі, завдячуючи технології клієнт/сервер;
- надійне функціонування серверу бази даних Firebird, який здатен працювати автономно;
- динамічне генерування простих веб - сторінок на основі даних, що утримуються в пакетах;

- формування звітних документів, з можливістю урахування вимог Міністерства освіти і науки України, у вигляді Microsoft Excel.

Найбільш корисними для автоматизації управління ВНЗ, є пакети підсистем «Колоквіум» та «Деканат» [3-4].

Пакет підсистеми «Колоквіум» слугує для комп'ютерного оцінювання знань студентів. Підсистема передбачає наступні функціональні можливості:

- зручна оболонка для формування тестів;
- тестування студентів у різних режимах та на основі результатів тестування оцінювати знання;
- забезпечується доступ до бази даних з результатами тестування з метою всебічного аналізу;
- автоматична генерація та друк звітів за результатами тестування;

Пакет підсистеми «Деканат» призначений для автоматизації планування та обліку навчального процесу в закладах освіти (технікумах, коледжах, училищах, інститутах та університетах). Пакет дозволяє створити та підтримувати базу даних, в якій формується та реєструється така інформація:

- структура навчального процесу закладу (спеціальності, групи, підгрупи, потоки, предмети, кількість годин з предметів за різними видами занять, навчальні плани та, при необхідності, факультети, кафедри, циклові комісії);
- дані щодо всіх викладачів закладу та їхнього планового навантаження, розклад роботи;
- щоденні дані про фактичну роботу кожного викладача з кожного предмету;
- великий обсяг даних щодо всіх студентів закладу та їхньої успішності за весь період навчання;
- дані про наявні корпуси та аудиторії навчального закладу, їхнього заповнення, розклад занять.

Головним недоліком пакетів, які представляє ПП «Політек-софт» є відсутність підтримки кредитно-модульної системи. Адже в рамках інтеграції України в європейське освітнє співтовариство проводиться поступова модернізація та удосконалення змісту вищої освіти та організації навчального процесу. Загальна кількість впроваджень програмного забезпечення ПП «Політек-СОФТ» в Україні на сьогодні складає 289 [5].

2.3. Програмний комплекс «АЛЬМА-МАТЕР»

Програмний комплекс «АЛЬМА-МАТЕР» є актуальним і своєчасним рішенням у сфері автоматизації навчального процесу. «Альма-Матер» дозволяє автоматизувати діяльність співробітників, що пов'язана з одержанням, обробкою, зберіганням й аналізом даних, необхідних для успішного функціонування навчального закладу. При розробці системи «Альма-Матер» використовувався кращий вітчизняний і закордонний досвід освітніх інформаційних проєктів. Програма може бути адаптована до структури будь-якого навчального закладу з різними видами й формами надання освітніх послуг [6].

Модулі «Альма-Матер»: «Приймальна комісія», «Деканат/навчальна частина», «Учбово-методичний відділ» (з можливістю формування розкладу).

Головні переваги:

1. Модульність. Можливість покупки програмного забезпечення частинами, а не цілим комплексом. Програмний комплекс допрацьовується під потреби кожного ВНЗ індивідуально.

2. Програмний комплекс розроблено відповідно до вимог Болонського процесу.

Програма «Альма Матер» написана мовою програмування PHP 5.2.1, база даних MySQL 5.0, Експорт й імпорт можливі з будь-яких програмних засобів і в будь-які інформаційні системи. Базова конфігурація серверу дозволяє

обслуговувати одночасно до 400 робочих місць. Кількість може бути розширена шляхом додавання виділеного серверу баз даних.

Функціональні можливості модулів:

1. Приймальна комісія:

- тестування абітурієнтів, автоматичне зарахування тих, що пройшли вступні тести;

- автоматичне занесення результатів поточних електронних тестів (іспитів/заліків) до бази даних;

- облік усіх абітурієнтів і студентів в єдиній базі даних.

2. Деканат/ учбова частина:

- облік успішності, рубіжний контроль;

- надання вільного доступу до інформації про відвідуваність і успішність студентів;

- облік особистих карток усіх студентів в єдиній базі;

- автоматизація введення екзаменаційно-залікових відомостей;

- автоматичне формування наказів на відрахування, списків груп, відомостей, особистих карток, журналів й інших стандартних документів як внутрішнього, так і зовнішнього документообігу;

- полегшення створення документів про рух студентів із збереженням історії про переміщення для подальшого аналізу;

- модуль обліку оплати за навчання, з можливістю установки індивідуальних розцінок для кожного студента і загальних по групах/потоках;

Головним недоліком програмного комплексу «АЛЬМА-МАТЕР», є довгий час на впровадження системи та постійна матеріальна підтримка вже впровадженої системи.

2.4 Висновки до Розділу 2

Одне з складних завдань, що стоять перед сучасним університетом, – це підвищення ефективності його роботи, а саме – комплексна дія на стратегію, мережеву інфраструктуру, організаційну структуру, системи управління і оплати, корпоративну культуру вищого навчального закладу.

Єдиним рішенням цієї задачі є автоматизація діяльності вузу і створення інформаційного середовища, яке з одного боку, надає доступ до необхідної користувачам актуальної, несуперечливої і повної інформації, з іншої – є необхідним інструментом діяльності співробітників вузу і навчання студентів.

Таке інформаційне середовище дозволяє управляти процесами, даними і людьми, тобто інформаційне середовище можна розглядати і з погляду підтримки життєдіяльності університету і як інструмент управління.

У результаті аналізу існуючих рішень було розглянуто велику кількість існуючих розв'язань задачі автоматизації навчального процесу. До уваги було прийнято функціональний набір програмних продуктів і їхні структури. Проте, незважаючи на велику кількість варіантів, доступних для використання, було зроблено висновок про їхню недостатність. Альтернативи якщо певною мірою і включають деякі необхідні функції, то жодна з альтернатив сама по собі не включає весь спектр необхідних функцій.

3. МЕТОДИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ

Для реалізації мобільного додатку була обрана операційна система Android. Операційна система Android - це операційна система, для сучасних мобільних пристроїв, що розроблена компанією Google, на основі модифікованої версії ядра Linux та іншого програмного забезпечення з відкритим вихідним кодом [8-9].

Android також асоціюється з комплектом фірмового програмного забезпечення, розробленого компанією Google, включаючи основні програми для таких послуг, як Gmail та Пошук Google, а також магазин прикладних програм і цифрова дистрибутивна платформа Google Play та відповідна платформа для розробників. Ці програми ліцензовані виробниками пристроїв Android, сертифікованими за стандартами, встановленими компанією Google, але AOSP було використано в якості основи конкуруючих екосистем Android, таких як Fire OS OC Amazon.com, які використовують власні еквіваленти цих Google Mobile Services.

Особливості розробки програмного комплексу для мобільного додатку:

- середовище розробки програмного забезпечення – Android Studio 3.4 (від Google);
- мова програмування – Kotlin;
- база даних була створена на основі сервісу Firebase(від Google);

Програмний комплекс працює на мобільних пристроях з операційною системою Android версій не нижче Android 5.0 Lollipop.

Мова Kotlin успішно прийнята великими компаніями, і деякі з них поділилися своїм досвідом: Pinterest успішно представив Kotlin у своєму додатку, який щомісяця використовує 150 мільйонів.

Тестування програмного комплексу виконано на емуляторі Android-смартфонів AVD (Android Virtual Device), а також на смартфоні з версією Android 5.0.

Для реалізації веб-додатку було використано мову TypeScript. TypeScript — це мова програмування, що позиціонується як засіб розробки веб-додатків, що розширює можливості стандартного JavaScript.

Особливості розробки програмного комплексу для веб-додатку:

- середовище розробки програмного забезпечення – Visual Studio Code;
- мова програмування – TypeScript;
- база даних була створена на основі сервісу Firebase(від Google).

3.1 Середовище розробки

Середовище розробки Android Studio — офіційне середовище розробки додатків для відповідної операційної системи, яке прийшло на зміну середовищу Eclipse, і є на момент найпопулярнішим інструментом серед Android розробників. Версія Android Studio 4.0 була випущена в грудні 2018 року [7]. Демонстрація інтерфейсу програмного середовища зображена на рисунку 3.1.

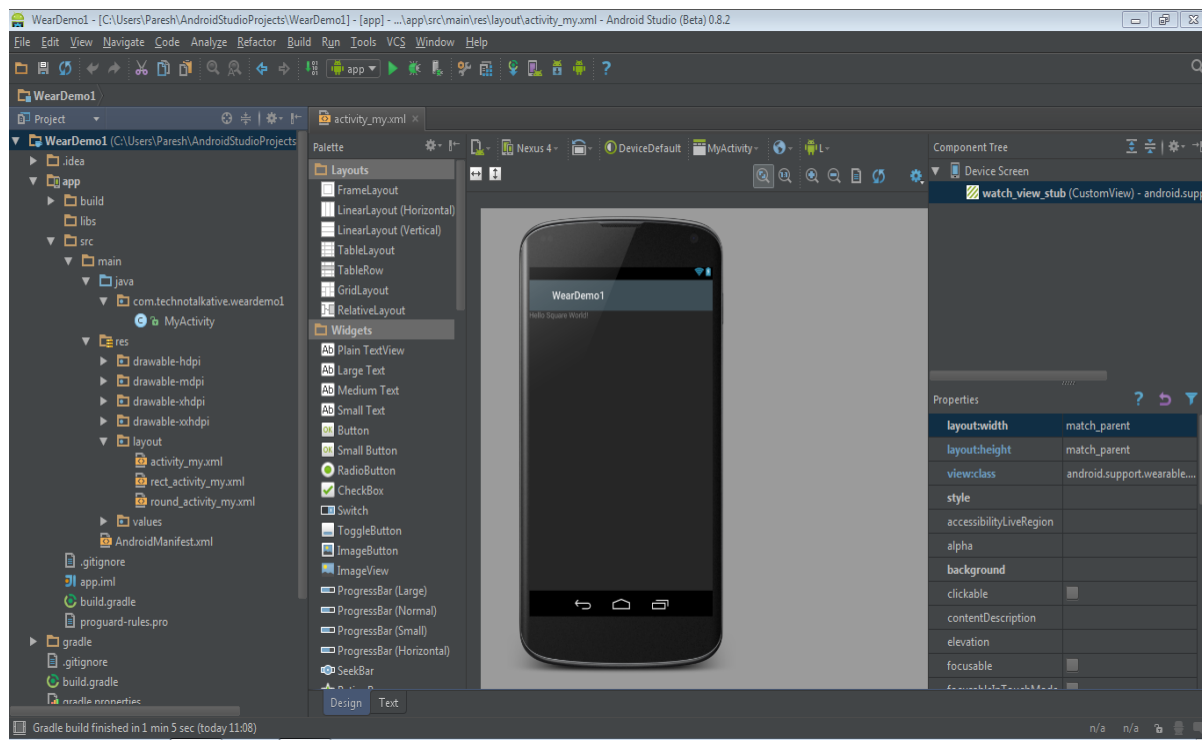


Рисунок 3.1. - Інтерфейс програмного середовища Android Studio.

Розробка в середовищі Android Studio є дуже гнучкою, оскільки середовище відображає всі робочі файли прямо в структурі проекту. Середовище адаптоване до виконання різних завдань. Також дане середовище має різні засоби для тестування програм.

Перевагами даного середовища розробки є наступні:

- серед підтримуваних мов програмування значаться C / C ++, Java і Kotlin;
- наявність зручного редактора коду;
- інструменти для швидкого редагування макета;
- є вбудований емулятор ОС Android AVD (Android Virtual Device), який може бути використаний для емуляції роботи планшета, телефона, пристроїв Android Wear і Android TV;
- наявність бібліотеки готових шаблонів і компонентів Android;
- інтеграція з хмарним сервісом Firebase Service;
- інтеграція з GitHub і Subversion (SVN);
- є вбудована програма для аналізу вмісту APK (APK Analyzer);
- наявність інструментів для тестування додатків, тощо.

Недоліками цього середовища є те, що потрібен потужний комп'ютер для запуску емулятора мобільного пристрою, а також середовище не підходить для написання серверних Java-проектів для ПК і Android.

3.2 Мова програмування Kotlin

Мова Kotlin - це універсальна статично типізована, з відкритим вихідним кодом, мова програмування для JVM і Android, яка поєднує в собі об'єктно-орієнтовані і функціональні можливості програмування. Вона сфокусований на оперативній сумісності, безпеці та простоті [10-12].

Отже, мова програмування Kotlin базується на JVM («Java Virtual Machine» - Віртуальна джава машина), тобто може бути декомпільована у Java код, тобто чудово підходить для розробки Android додатків, приносить всі переваги

сучасної мови для платформи Android без введення нових обмежень.

Слід розглянути декілька важливих пунктів цієї мови:

Сумісність: мова програмування Kotlin повністю сумісна з JDK 6, а це забезпечує те, що додатки на мові Kotlin можуть працювати на старих пристроях Android без проблем. Інструмент Kotlin повністю підтримується в Android Studio і сумісний з системою Android build.

Продуктивність: додаток Kotlin працює так само швидко, як еквівалентний Java, завдяки дуже схожим структурам байт-кодів. Kotlin має багато сучасних вбудованих функцій програмування, як от лямбда-операції, програмний код, що написаний за їх допомогою часто виконується навіть швидше, ніж той же код, написаний на Java.

Оперативна сумісність: мова Kotlin є 100% сумісним з Java, що дозволяє використовувати всі існуючі бібліотеки Android у програмі Kotlin. Це включає в себе обробку анотацій, що дозволяє використовувати бібліотеки для ін'єкції залежностей.

Діяльність: мова Kotlin має дуже компактну бібліотеку, яка може бути зменшена завдяки використанню ProGuard. У реальному додатку тривалість роботи Kotlin додає лише кілька сотень методів і менше 100K до розміру файлу .apk.

Час компіляції: мова Kotlin підтримує ефективну комбіновану поетапну компіляцію, так що при наявності додаткових накладних витрат на чисті збірки додаткові збірки зазвичай стають швидкими та швидшими, ніж з Java.

Світовий досвід: мова Kotlin офіційно підтримується Google. Google офіційно рекомендує використовувати мову Kotlin, як основну мову програмування для операційної системи Android.

Інтеграція Kotlin з Android: розширення для Android Kotlin - розширення компілятора, яке дозволяє позбутися від викликів findViewById() у коді та замінити їх синтетичними властивостями, створеними компілятором.

Anko - бібліотека, яка надає набір додаткові класи-обгортки навколо API-інтерфейсів Android, а також DSL, що дозволяє замінити файли формату .xml.

3.3 Бібліотека ін'єкції залежностей Dagger

3.3.1. Основна робота принципу ін'єкції залежностей

В інженерії програмного забезпечення, ін'єкція залежностей є методом, за допомогою якого один об'єкт (або статичний метод) постачає залежності іншого об'єкта. Залежність - це об'єкт, який може бути використаний (послуга). Ін'єкція - це проходження залежності до залежного об'єкта (клієнта), який його використовуватиме. Сервіс стає частиною клієнта. Передача сервісу клієнту, а не надання клієнту можливості для створення або пошуку послуги, є основною вимогою моделі [13-14].

Ця основна вимога означає, що використання значень (послуг), вироблених у класі з нових або статичних методів, заборонено. Клієнт повинен прийняти залежності, передані ззовні. Намір ін'єкції залежностей полягає в тому, щоб відокремити об'єкти в тій мірі, в якій ніякого клієнтського коду не потрібно змінювати, просто тому, що об'єкт, від якого залежить, повинен бути змінений на інший.

Залежність ін'єкцій є однією з форм більш широкої методики інверсії контролю. Як і в інших формах інверсії керування, інверсія залежить від принципу інверсії залежностей. Клієнт делегує відповідальність за надання своїх залежностей зовнішньому коду (інжектор). Клієнт не може викликати код інжектора, код ін'єкції, який конструює послуги та закликає клієнта ввести залежності. Це означає, що клієнтський код не повинен знати про код ін'єкції, про те, як побудувати служби або навіть про ті фактичні послуги, які він використовує; клієнт повинен лише знати про внутрішні інтерфейси служб, оскільки вони визначають способи використання клієнтами залежностей. Це відокремлює обов'язки щодо використання та створення залежностей.

Існує три поширені способи для клієнта приймати ін'єкції залежності: інтеракція (сеттер), інтерфейс та ін'єкція на основі конструктора. Ін'єкції сеттера

та конструктора головним чином відрізняються способом використання та доцільністю використання в тій чи іншій ситуації. Ін'єкція інтерфейсу відрізняється тим, що залежність дає шанс контролювати власну ін'єкцію. Кожен метод вимагає, щоб окремий будівельний код (інжектор) ніс відповідальність за ініціалізацію клієнта та його взаємозв'язки залежностей між собою.

3.3.2 Використання Dagger

Бібліотека Dagger - це повністю статична, залежна від часу компіляції бібліотека, що підходить як для Java, так і для Android. Dagger - це заміна фабричних класів, яка реалізує схему дизайну ін'єкцій залежностей, що не потребує попереднього написання власної реалізації. Це дозволяє зекономити більше часу на проектування та розробку системи. Необхідний принцип роботи-оголосити залежності, які мають постачатися, вказати вихідний тип кожної залежності будь- яким способом [15-16].

Dagger будує екземпляри класів додатку і ініціалізує їх залежності. Він використовує `javax.inject.Inject` анотацію, щоб визначити, які конструктори та поля необхідно проініціалізувати. Слід використовувати анотацію `@Inject`, щоб анотувати конструктор, який Dagger повинен використати для створення екземпляру класу. Коли запитується новий екземпляр, Dagger отримає необхідні значення параметрів і використає цей конструктор.

Якщо у вашому класі є `@Inject` анотовані поля, але немає `@Inject` анотованого конструктора, то Dagger буде вводити ці поля за запитом, але не створюватиме нові екземпляри. Додайте конструктор без аргументів з анотацією `@Inject`, щоб вказати, що Dagger також може створювати екземпляри.

Бібліотека Dagger також підтримує і спосіб метод ін'єкції, хоча ін'єкція через конструктор або ін'єкція через анотоване поле, як правило, є кращими. Класи, в яких відсутні атрибути `@Inject`, не можуть бути побудовані за допомогою цієї бібліотеки.

За замовчуванням, Dagger ініціалізує кожну залежність, будуючи екземпляр запитуваного типу, як описано вище. Коли ви запитуєте певний тип, скажімо тип об'єкту `DocumentMaker()`, він отримає його, викликавши `new DocumentMaker ()` та встановивши його ін'єкційні поля.

Але слід пам'ятати, що `@Inject` не працює скрізь:

- Інтерфейси не можуть бути проініціалізовані.;
- зовнішні класи не можуть бути анотовані;
- настроювані об'єкти повинні бути налаштовані.

Для таких випадків, коли `@Inject` недостатньо або його доволі незручно використовувати – використовуйте анотований метод `@Provides`, щоб ініціалізувати залежність. Тип значення, що повертає метод визначає, яку залежність анотація задовольняє. Анотовані методи `@Inject` і `@Provides` утворюють граф об'єктів, пов'язаних між собою своїми залежностями. Код виклику ініціалізації графу залежностей має знаходитись в батьківському(початковому) класі програми, як от клас `Application` для Android.

3.4 Бібліотека ReactiveX

Бібліотека ReactiveX - це бібліотека для створення асинхронних програм і подій за допомогою спостережуваних послідовностей [17-18].

Бібліотека розширює схему спостереження для підтримки послідовностей даних та/або подій і додає оператори, що дозволяють складати послідовності подій декларативно, а також дозволяє абстрагуватися від побоювань щодо таких речей, як вирішення низькорівневих задач, що пов'язані з потоками, їх синхронізацією, безпекою, паралельною структурою даних та оброблення нестандартних даних, блокуванням вводу-виводу.

Спостерігачі (згідно з патерном програмування «Observable») заповнюють прогалини, будучи ідеальним способом доступу до асинхронних послідовностей кількох елементів. Часом такий стиль програмування називають "функціональним реактивним програмуванням", але це неправильно. ReactiveX

може бути функціональним, і він може бути реактивним, але "функціональне реактивне програмування" - це інша дещо інше. Одним з основних моментів відмінності є те, що функціональне реактивне програмування діє на значення, які постійно змінюються у часі, тоді як ReactiveX працює за дискретними значеннями, які надходять з часом.

Модель ReactiveX Observable дозволяє обробляти потоки асинхронних подій за допомогою тих самих простих/складних операцій, які використовуються для збірок елементів даних, таких як масиви. Це звільняє від заплутаних мереж зворотних викликів, і тим самим ця функціональність робить код більш читабельним і менш схильним до помилок.

Готові можливості, які надає Java- Futures, це проста для використання техніка, що знаходиться на одному рівні асинхронного виконання, але ця реалізація починає додавати досить нетривіальні складності, коли її структура ускладнюється, до прикладу вкладені Futures.

Важко використовувати техніку Futures для оптимального складання умовних асинхронних потоків виконання (або неможливо, оскільки затримки кожного запиту змінюються під час виконання). Це, звичайно, може бути зроблено, але реалізація значно ускладнюється, або передчасно блокує Future.get (), що виключає переваги асинхронного виконання. З іншого боку, ReactiveX Observables призначені для створення потоків та послідовностей асинхронних даних.

3.4.1 Гнучкі Observables

ReactiveX спостерігачі підтримують не тільки виділення одиничних скалярних значень (як це робить Futures), але й послідовності значень чи навіть нескінченних потоків. Observable - одна абстракція, яку можна використовувати для будь-якого з випадків використання. Спостерігач має всю гнучкість та елегантність, пов'язану зі своїм дзеркальним відображенням – Iterable (рис. 3.2).

An Observable is the asynchronous/push “dual” to the synchronous/pull Iterable

event	Iterable (pull)	Observable (push)
retrieve data	<code>T next()</code>	<code>onNext(T)</code>
discover error	throws <code>Exception</code>	<code>onError(Exception)</code>
complete	<code>!hasNext()</code>	<code>onCompleted()</code>

Рисунок 3.2 - Відмінність функціонування Observable та Iterable

Бібліотека ReactiveX надає сукупність операторів, за допомогою яких ви можете фільтрувати, вибирати, перетворювати, об'єднувати і створювати Observables. Це дозволяє ефективно виконувати та складати об'єкти, як результати виконуваних операцій. Необхідно розуміти, що клас Observable може виступати як евівалент до Iterable, але з різними значеннями, оскільки Observable обробляє надіслані запити, а Iterable лише повертає результат. За допомогою властивості Iterable- споживача, він отримує значення від виробника та блокує основний потік вводу до тих пір, поки значення не з'являться. Цей підхід є більш гнучким, оскільки значення можуть надходити синхронно або асинхронно.

3.4.2 Використання бібліотек RxJava та RxJs

Бібліотеки RxJava та RxJs - бібліотеки для створення асинхронних програм на основі подій за допомогою спостережуваних послідовностей.

Кожна з бібліотек розширює патерн програмування «Observable», для підтримки послідовностей даних/подій і надає додаткові оператори, які дозволяють складати послідовності разом декларативно, а також абстрагують таких речей, як низькорівневі потоки, синхронізація даних, безпека потоків та паралельна обробка структур даних.

RxJava підтримує мови Java 6 або вище та на базі JVM, такі як Groovy, Clojure, JRuby, Kotlin і Scala. RxJava призначена для більшої кількості середовищ

ніж просто Java / Scala, і розробляється з урахуванням ідіом кожного мови на базі JVM. (Це те, що ми все ще працюємо.)

У RxJava та RxJs існує об'єкт, який реалізує інтерфейс Observer, підписується на об'єкт класу Observable. Потім цей об'єкт реагує на будь-який об'єкт або послідовність елементів, який випромінює об'єкт Observable. Ця схема полегшує виконання паралельних операцій, оскільки вона не потребує блокування під час очікування, поки Observable виділятиме об'єкти, а замість цього створить вартувий у вигляді підписника, який готовий належним чином реагувати на будь-який майбутній час, який Observable робить.

Надамо чітке значення одного з головних підкласів ConnectableObservable та розглянемо найголовніші його оператори, які використовувались в роботі:

`ConnectableObservable.connect()` - доручає Connectable Observable почати випромінювати елементи.

`Observable.publish()` – звичайний Observable, але при підписанні на нього він не починає випускати елементи, але починає робити це тільки коли до нього застосовується оператор `Connect`. Таким чином, існує можливість підказати Observable почати випускати елементи в потрібний момент або ж за настання певної умови. Візуальна робота методу зображена на рис. 3.3.

`Observable.replay ()` - гарантує, що всі підписники побачать таку ж послідовність випущених елементів, навіть якщо вони підписалися після того, як Observable почав випускати елементи.

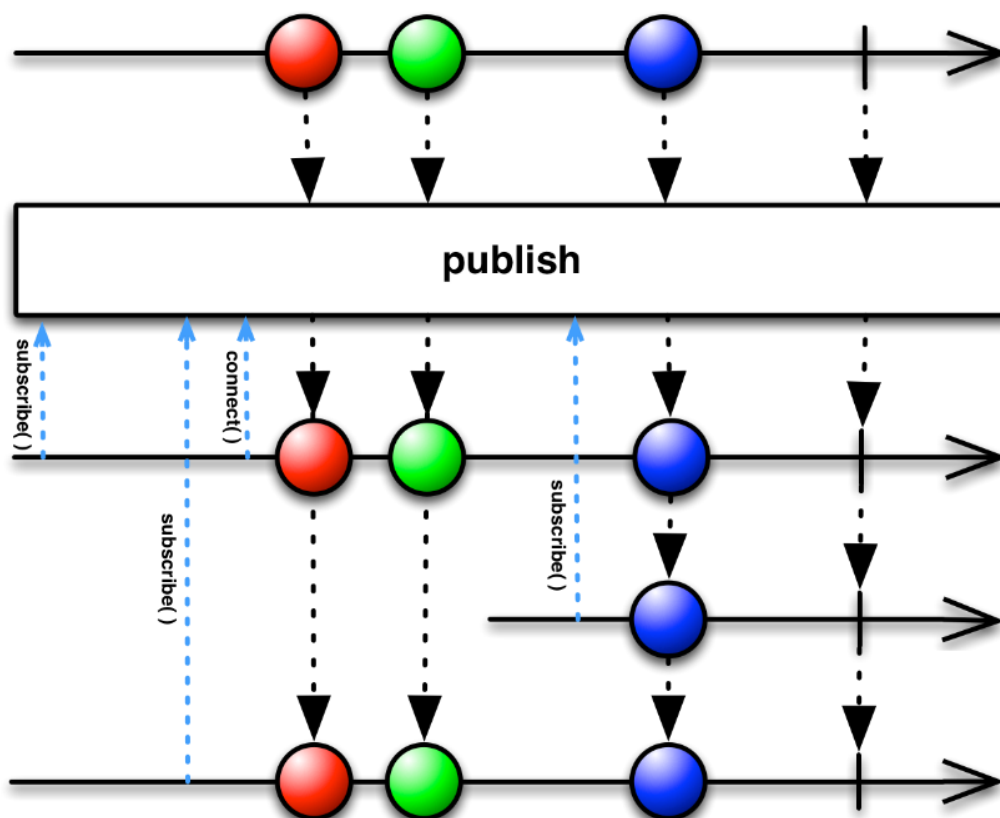


Рисунок 3.3 - Ланцюжкова подія у бібліотеці

3.4.3 Використання бібліотеки Retrofit

Retrofit - це клієнт REST для Android і Java. Це дозволяє відносно легко скачувати і завантажувати JSON (або інші структуровані дані) через веб-сервіс на основі REST. У Retrofit є можливість налаштувати конвертер, що буде використовуватися для серіалізації даних. Зазвичай для JSON використовується GSON, але можна додати власні конвертери для обробки XML або інших протоколів.

Retrofit використовує бібліотеку OkHttp для HTTP-запитів. Для роботи з Retrofit потрібно в основному три класи.

- Клас моделі, що використовується для зіставлення даних JSON;
- Інтерфейси, що визначають можливі HTTP операції;
- Модернізація. Клас Builder - примірник, що використовує інтерфейс API Builder, який дозволяє визначити кінцеву точку URL HTTP операції.

Кожен метод інтерфейсу представляє один можливий виклик API. Він повинен мати адресу http для того щоб вказати тип запиту і відносний URL-адресу. Результат переноситься у результуюче значення на об'єкт виклику з типом очікуваного результату.

Приклад найпростішої реалізації зображено на рис. 3.4:

```
@GET("users")
```

```
Call<List<User>> getUsers()
```

Рисунок 3.4 - Демонстрація методу, що запитує список користувачів

3.5 Опис програмного використання платформи Firestore

Firestore - це гнучка, масштабована хмарна база даних від Firebase і Google Cloud Platform для веб-додатків, мобільних платформ, і серверних додатків. Як і Firebase Realtime Database, вона синхронізує дані між клієнтськими додатками за допомогою слухачів реального часу а також пропонує підтримку оффлайн режиму для мобільних платформ і веб-додатків [19-20].

Модель даних Firestore підтримує гнучкі, ієрархічні структури даних. Зберігає дані в документах, які в свою чергу зберігаються в колекціях. Документи можуть мати вкладення об'єкти і підколекції.

У Firestore, користувач може робити запити для отримання одного документа, певного шуканого документа, або всіх документів в колекції, які відповідають параметрам запиту. Firestore кешує використовувані додатками дані, з допомогою цього додаток може писати, читати, слухати оновлення і робити запити навіть коли девайс знаходиться в офлайн. Коли девайс повернеться в онлайн, то Firestore порівняє зміни між локальним сховищем та даними у хмарі, та оновить відмінні дані.

Firestore - це збережена в хмарі, NoSQL база даних, з якої iOS, Android і веб-додатки працюють безпосередньо з допомогою нативних SDK. Крім того

доступні SDK для Node.js, Java, Python, Go SDK, а також REST api.

3.6 Висновки до Розділу 3

У розділі розглянуто опис програмного забезпечення з допомогою якого можливо реалізувати систему автоматизації та аналізу навчального процесу. Обрані технології реалізації є передовими та користуються великим попитом на ринку праці, програмні комплекси, що є продуктом їх використання є надійними, легко підтримуються та оновлюються.

Створені веб-додаток та мобільний додаток містять сервіси та компоненти для роботи з віддаленою базою даних Firestore, реалізують асинхронну роботу за допомогою сучасної бібліотеки Rx та реалізують такий шаблон програмування, як Dependency Injection.

4. ОПИС ФУНКЦІОНАЛУ СИСТЕМИ ЗА ОСНОВНИМИ ТИПАМИ КОРИСТУВАЧІВ

Реалізація будь-якої системи перш за все потребує аналізу і пошуку оптимального шляху вирішення поставленого завдання. Після ретельного аналізу поставленого завдання було виокремлено основні критерії, яким має відповідати система з програмної точки зору, а саме:

- незалежність розроблених частин програмного продукту, для полегшення написання майбутнього функціоналу;
- можливість розширення функціоналу кінцевого користувача з мінімальними змінами користувацького інтерфейсу.

Нижче розглянуто діаграму прецедентів (рис.4.1). Діаграма прецедентів - це динамічна діаграма або схема поведінки, що зображена в UML. Діаграму прецедентів слід використовувати при моделюванні функціональності системи за допомогою акторів та їх функціональних можливостей.

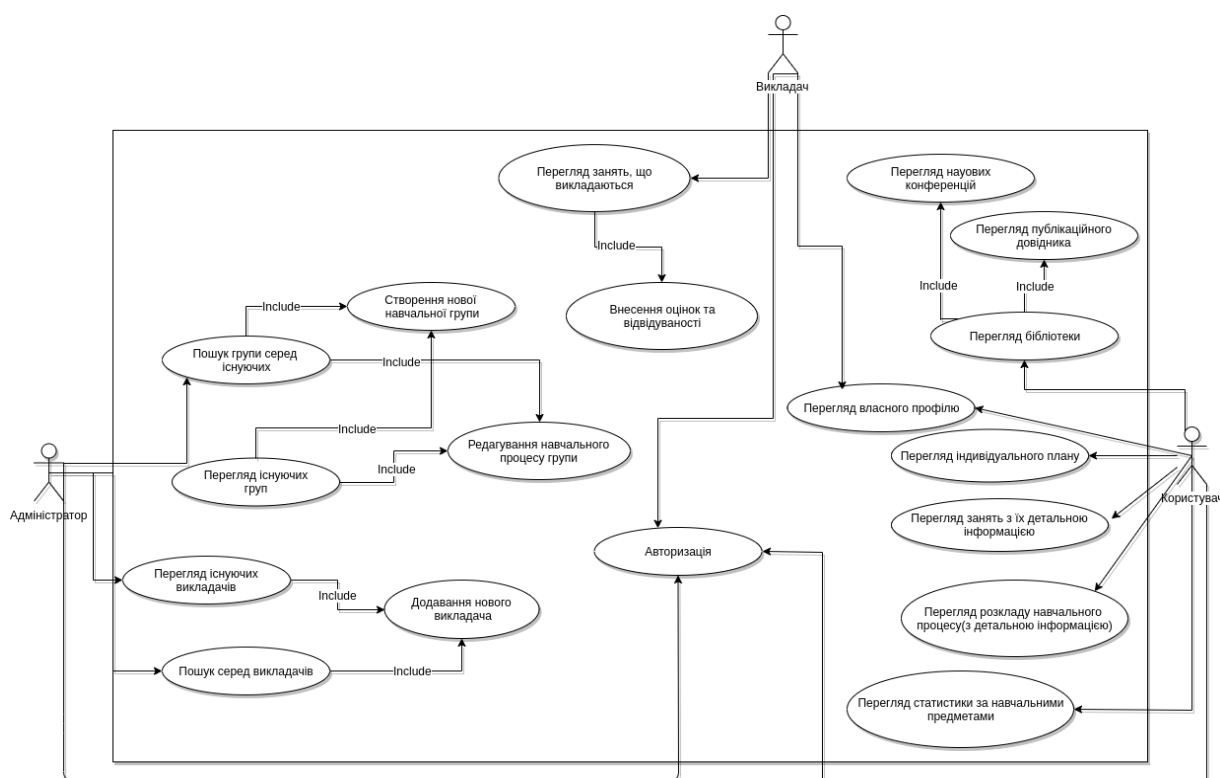


Рисунок 4.1 - Діаграма прецедентів системи

4.1. Опис функціоналу системи з точки зору адміністратора

Адміністратор має наступні функціональні можливості (рис.4.2):

- визначення певних ролей для користувачів, а саме: адміністратор, викладач та користувач (аспірант);
- авторизація у системі у якості адміністратора;
- пошук серед існуючих у системі навчальних груп;
- створення нових навчальних груп;
- редагування навчального процесу обраної групи (додавання навчальних предметів у обрану групу та призначення викладача для кожного навчального предмету);
- додавання нових викладачів до системи;
- пошук викладачів серед вже існуючих;
- реєстрація нового викладача у системі.

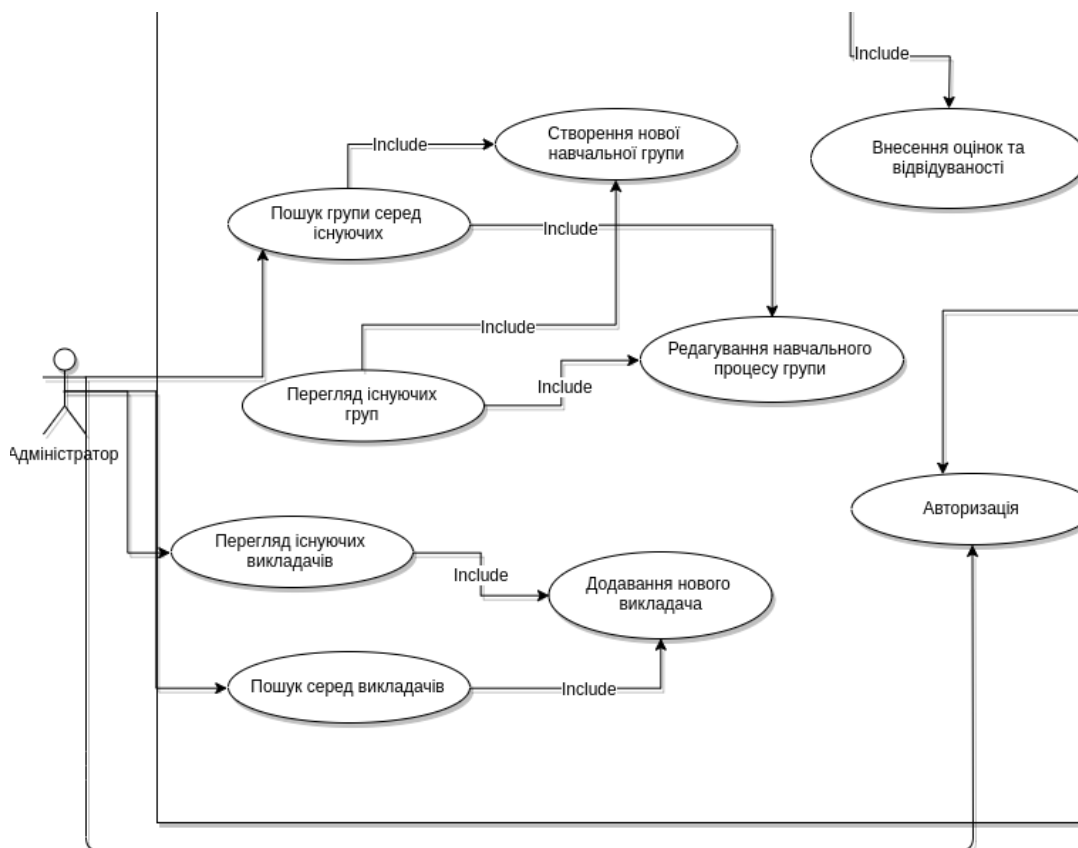


Рисунок 4.2 - Діаграма прецедентів для ролі адміністратора

4.2. Опис функціоналу системи з точки зору викладача

Викладач має наступні функціональні можливості (рис.4.3):

- перегляд сторінки власного профілю;
- авторизація у системі;
- перегляд занять та груп, де викладає викладач;
- виставлення оцінок/пропусків занять студентам за певними датами занять, що навчаються у групі, де викладає викладач;

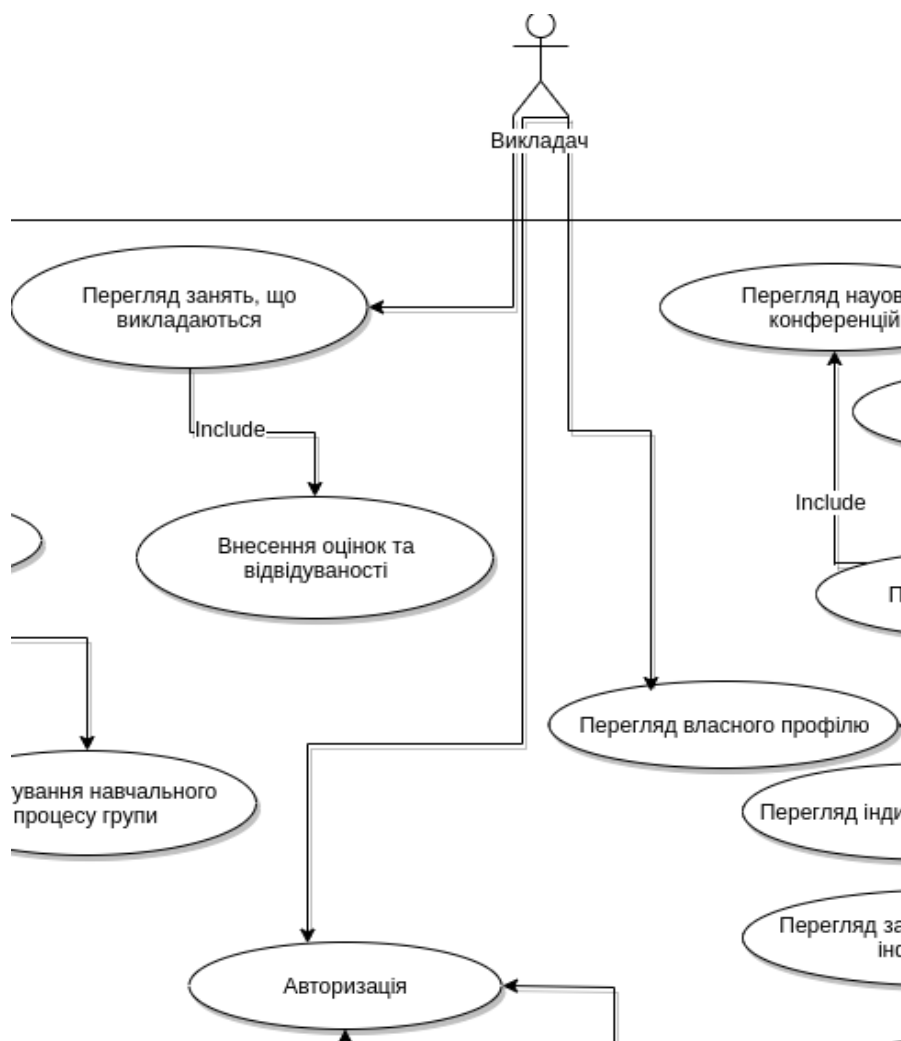


Рисунок 4.3 - Діаграма прецедентів для ролі викладача

4.3. Опис функціоналу системи з точки зору користувача (аспіранта)

Користувачем системи виступає аспірант. Саме для аспіранта було додатково створено Android додаток, для надання актуальної інформації у будь-якому місці у будь-який час.

Користувач має наступні функціональні можливості (рис.4.4):

- перегляд власного профілю;
- перегляд індивідуального плану;
- перегляд занять (оцінок та присутності) за певним навчальним предметом;
- перегляд розкладу занять;
- перегляд статистики за власною успішністю;
- перегляд поточних та майбутніх наукових конференцій;
- доступ до публікаційного довідника;
- доступ до мобільного додатку на операційній системі Android, що матиме всі вищеописані функції для ролі користувача.

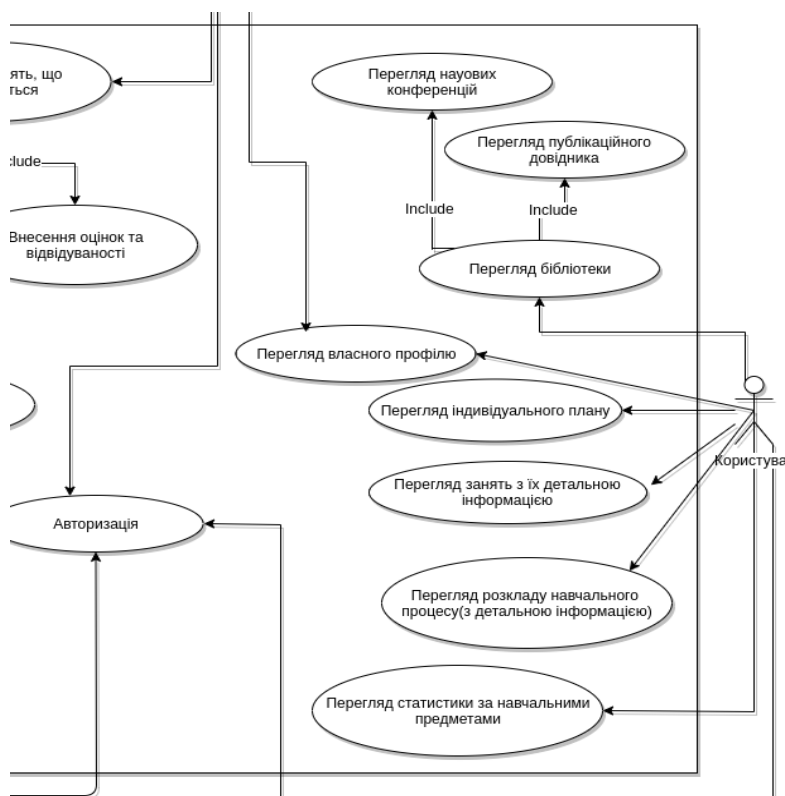


Рисунок 4.4 - Діаграма прецедентів для ролі аспіранта

4.4. Приклад взаємодії з системою

Далі розглянемо діаграму послідовності, на прикладі користувача (рис.4.5).

Діаграма послідовностей відображає взаємодії об'єктів, що розташовані у певній часовій послідовності.

Як можна побачити, при відкритті додатку відбувається перевірка наявності даних користувача, якщо дані існують, то відбувається авторизація користувача у додатку, якщо ні, то користувач автоматично переходить на екран авторизації. Після вдалої авторизації відбувається завантаження даних користувача та відображення його інформаційного профілю, після цього користувач вже має доступ до всіх вищеописаних можливостей.

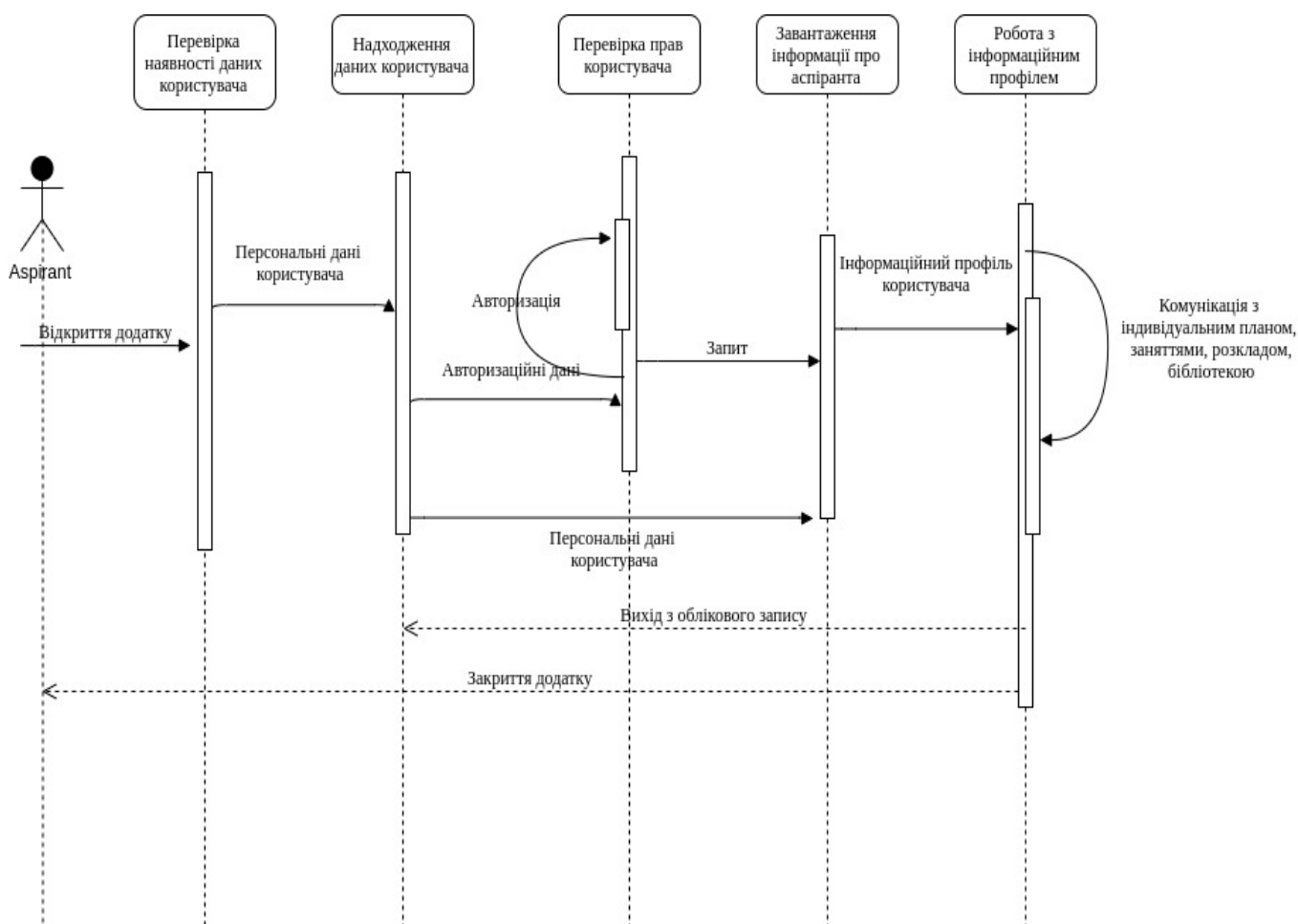


Рисунок 4.4 - Діаграма послідовності використання додатку

4.5 Висновки до розділу 4

У цьому розділі на основі попередніх пунктів було знайдено оптимальні шляхи для реалізації системи, виокремлено основні критерії, яким мають відповідати програмні модулі системи. Система реалізує три ролі користувачів у системі. Для кожної з ролей було поставлено функціональні вимоги, що відображаються на діаграмах прецедентів. Крім того було побудовано діаграму послідовності, що відображає покрокову взаємодію користувачів з веб-додатком або Android додатком.

5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ

Розроблений програмний компонент складається з трьох модулів, а саме: модулю адміністратора, модулю викладача та модулю студента. Програмні модулі реалізовані як веб додатки. Слід зазначити, що модуль студента додатково реалізовано у вигляді програмного застосунку для операційної системи Android.

5.1. Опис спільних частин при взаємодії з програмним компонентом

Програмний компонент реалізує три незалежні модулі. Спільною частиною для кожного з трьох модулів є підмодуль авторизації користувача. Він дозволяє авторизуватися у системі, ввівши логін та пароль. Підмодуль авторизації з'являється автоматично при відкритті веб додатку або Android додатку вперше, у вигляді діалогового вікна. З допомогою логіну та паролю підмодуль автоматично визначає роль користувача у системі, саме він розмежовує логіку використання програмних модулів за ролями користувачів.

Нижче проілюстровано вікно авторизації користувача (рис.5.1), з тестовою поштовою адресою у пункті вводу «Email» та тестови паролем, у пункті вводу «Пароль». Слід зауважити, що при закінченні програмної сесії користувачем, система автоматично запам'ятовує останнього авторизованого користувача та при наступній його роботі з додатком автоматично авторизує. Інструмент виходу, вилогіну, з системи буде показано у пункті нижче.

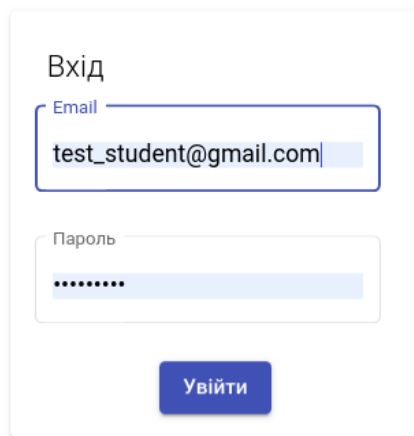


Рисунок 5.1 - Вікно авторизації

5.2. Опис взаємодії з модулем студента

Модуль студента реалізовано як у вигляді веб-додатку, так і у вигляді Android додатку.

Після успішної авторизації у системі, з допомогою підмодуля авторизації, користувач автоматично перенаправляється на сторінку власного профіля, що містить наступну інформацію (рис.5.2 (а,б)):

- ПІБ, що означає прізвище, ім'я, по-батькові користувача;
- Інститут/Факультет, що відображає скорочену абривіатуру інституту або факультету, де навчається користувач;
- Кафедра, що відображає назву кафедри, на факультеті, де навчається користувач;
- Спеціальність, що відображає номер спеціальності на кафедрі навчання користувача;
- Науковий керівник, відображає ім'я обраного наукового керівника;
- Термін підготовки;
- Рік вступу;
- Рік закінчення;
- Група, відображає абривіатуру навчальної групи користувача;

- Форму підготовки, відображає форму підготовки користувача, може мати лише три форми, а саме: денну, вечірню та заочну.

Аспірантура КПІ	
Мій профіль	ПІБ: Іванов Іван Іванович
Індивідуальний план	Інститут/Факультет: ТЕФ
Заняття	Кафедра: АПЕПС
Розклад	Спеціальність: 122
Статистика	Науковий керівник: Михайленко М. А.
	Термін підготовки: 3 роки
	Рік вступу: 2019
	Рік закінчення: 2022
	Група: тр-91ф
	Форма підготовки: Денна

Рисунок 5.2 - Сторінка профілю (а)

Профіль	
Ім'я:	Іванов Іван Іванович
Інститут/факультет:	ТЕФ
Кафедра:	АПЕПС
Спеціальність:	122
Група:	ТР-91Ф
Науковий керівник:	Михайленко М. А.
Рік вступу - рік закінчення:	2019 - 2022 роки
Форма підготовки:	Денна

Рисунок 5.2 - Сторінка профілю (б)

Зі сторінки власного профілю користувач може переходити до інших сторінок системи через меню, що знаходиться з ліворуч.

Одним з підпунктів меню є пункт «Індивідуальний план» (рис.5.3 (а,б)).

Сторінка індивідуального плану відображає навчальні дисципліни, що викладаються у навчальному семестрі. Кожна навчальна дисципліна має кількість кредитів та форму атестації, що встановлюються адміністратором системи.

Мій профіль	Індивідуальний навчальний план		
Індивідуальний план			
Заняття			
Розклад			
Статистика			
	1 семестр		
	Назва навчальної дисципліни	Кількість кредитів	Форма атестації
	Іноземна мова для наукової діяльності 1	8	Екзамен
	Екзистенціальна філософія 1. Практична риторика	8	Екзамен
	2 семестр		

Рисунок 5.3 - Сторінка індивідуального плану (а)

00:52

Індивідуальний навчальний план

1 семестр

Назва навчальної дисципліни:

Екзистенціальна філософія 1. Практична риторика

Кількість кредитів: 4

Форма атестації: Екзамен

Назва навчальної дисципліни:

Іноземна мова для наукової діяльності 1

Кількість кредитів: 4

Форма атестації: Екзамен

Рисунок 5.3 - Сторінка індивідуального плану (б)

Наступним пунктом навігаційного меню є пункт «Заняття» (рис.5.4 (а,б)). Він відображає поточний курс навчальних дисциплін користувача та додаткову інформацію у вигляді відміток по проведеному занятті викладачем. Відмітки відображаються у вигляді строк, де на початку строки знаходиться дата заняття, а потім і підсумок заняття. Підсумки можуть відображатися у вигляді оцінок, наприклад «отримав 4», або «отримав 5», або, якщо студент, не з'явився на заняття у вигляді відмітки про пропуск «був відсутній». Підсумок по кожному заняттю має можливість додавати лише викладач даного заняття.

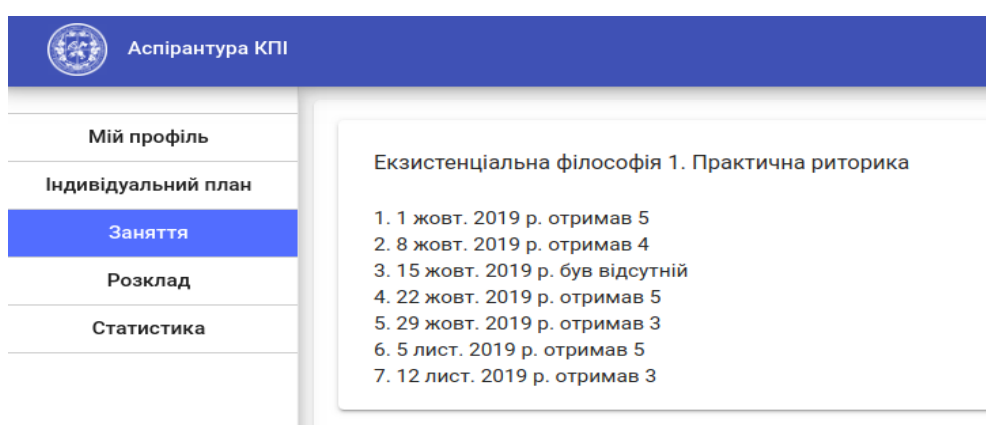


Рисунок 5.4 - Пункт «Заняття» (а)

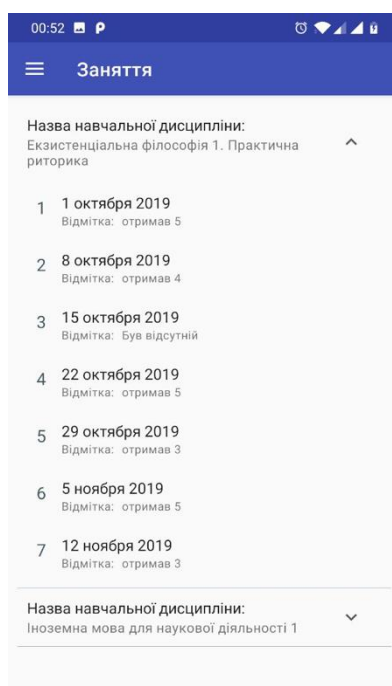


Рисунок 5.4 - Пункт - «Заняття» (б)

Наступним пунктом, після пункту «Заняття» є пункт «Розклад» (5.5 (а,б)).

Пункт «Розклад» відображає графік учбового процесу користувача, він є інтегрованим модулем, тому що має зв'язок з зовнішнім сервісом «Розклад КПП», та отримує всю важливу інформацію про заняття, а саме: номер пари, назву дисципліни, час проведення пари, номер навчального корпусу та номер аудиторії, тип навчальної дисципліни.

Мій профіль	I тиждень					
	Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота
Заняття						
Розклад	1			Іноземна мова для наукової діяльності 1 519-19 Прак доц. Олізко Ю. М.		
Статистика	2			Іноземна мова для наукової діяльності 1 519-19 Прак доц. Олізко Ю. М.		
	3			Екзистенціальна філософія 1. Практична риторика 110-07 Лек проф. Препотенська М. П.		
	4			Екзистенціальна філософія 1. Практична риторика 524-19 Прак доц. Покуліта І. К.		
	5					

Рисунок 5.5 - Пункт «Розклад» (а)

00:52

ТР-91Ф

— Перший тиждень —

Четвер

1	Іноземна мова для наукової діяльності 1	08:30 - 10:05 519-19 Прак
2	Іноземна мова для наукової діяльності 1	10:25 - 12:00 519-19 Прак
3	Екзистенціальна філософія 1. Практична риторика	12:20 - 13:55 110-07 Лек
4	Екзистенціальна філософія 1. Практична риторика	14:15 - 15:50 524-19 Прак

— Другий тиждень —

Четвер

1	Іноземна мова для наукової діяльності 1	08:30 - 10:05 519-19 Прак
4	Екзистенціальна філософія 1. Практична риторика	14:15 - 15:50 524-19 Прак

Рисунок 5.5 - Пункт «Розклад» (б)

По натисканню на пункт розкладу з мобільного додатку, користувач може перейти до детального опису обраної пари (рис.5.6). Детальна інформація додатково відображає повне ім'я викладача та карту з місцезорозташуванням пари. При натисканні на карту користувач переходить до зовнішнього додатку з мапою.

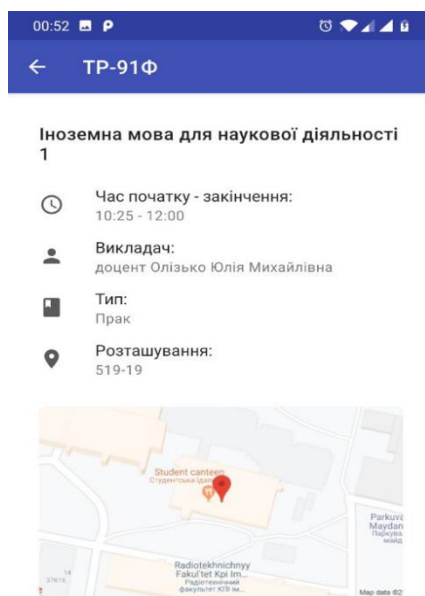


Рисунок 5.6 - Детальний опис обраної пари

Останній пункт - сторінка «Статистики» (рис.5.7). Відображає статистику успішності користувача у рамках предметів. Статистика ілюструється за допомогою графіків, та містить дані всіх відміток користувача зі сторінки «Заняття».

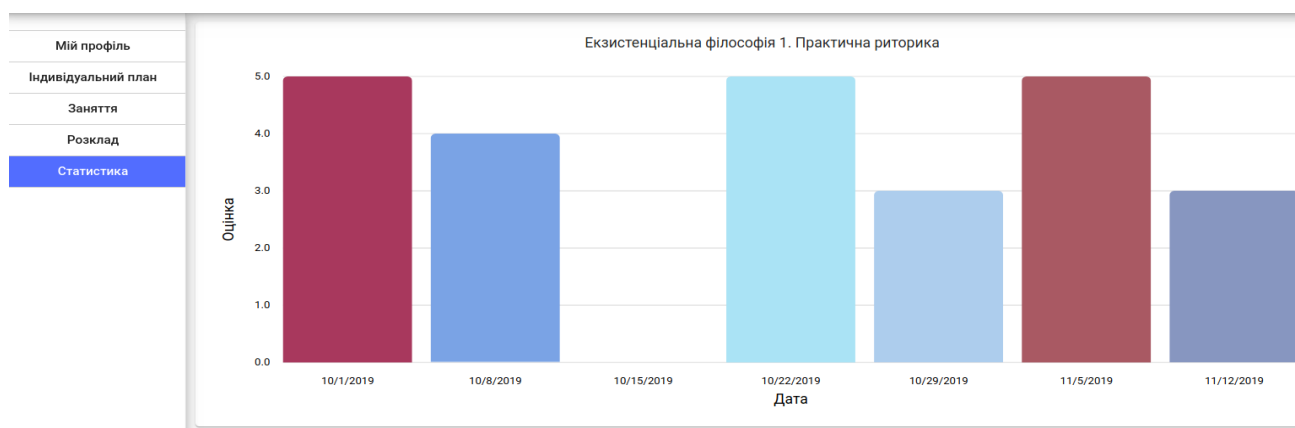


Рисунок 5.7 - Сторінка «Статистики»

5.3. Опис взаємодії з модулем викладача

Модуль викладача реалізовано у вигляді веб-додатку. Після успішної авторизації викладача перенаправляє до сторінки власного профілю, що містить коротку особисту інформацію, а саме: ПІБ, факультет реєстрації викладача та кафедру реєстрації користувача (рис.5.8).

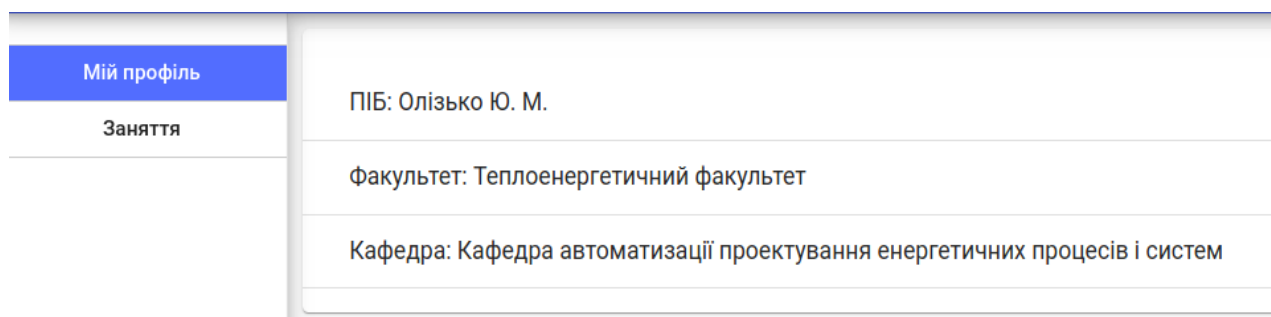
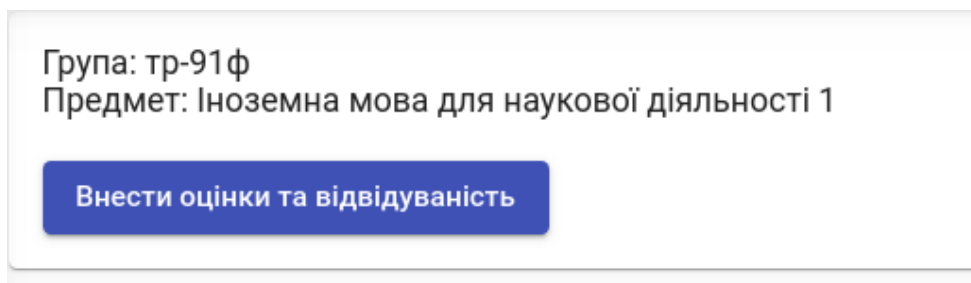


Рисунок 5.8 - Сторінка «Статистики»

У навігаційній панелі, що знаходиться ліворуч від основного вікна викладач має можливість перейти до пункту «Заняття».

Пункт «Заняття» відображає список, що складається з карток груп, де викладає викладач. Кожен елемент списку містить інформацію про назву групи, предмет, що викладається, та кнопку внесення оцінки та відвідуваності. По натисканню на кнопку внесення оцінки та відвідуваності, картка відповідної групи розгортається та відображає список студентів в групі. По натисканню на елемент списку, відповідна картка зі студентом розгортається та відображає інформацію про успішність студента (рис.5.8). Внизу інформаційної картки студента розміщується кнопка «Додати відмітку» (рис.5.9). При натисканні на кнопку з'являється діалогове вікно, що слугує для введення атестаційного підсумку за заняття відповідного студента (рис.5.10). Діалогове вікно дозволяє внести інформацію про: дату проведення заняття, додати відмітку за заняття або позначити заняття, як пропущене. Внизу діалогово вікна знаходяться дві кнопки: «Закрити» та «Додати». Кнопка «Закрити» закриває діалогове вікно, не

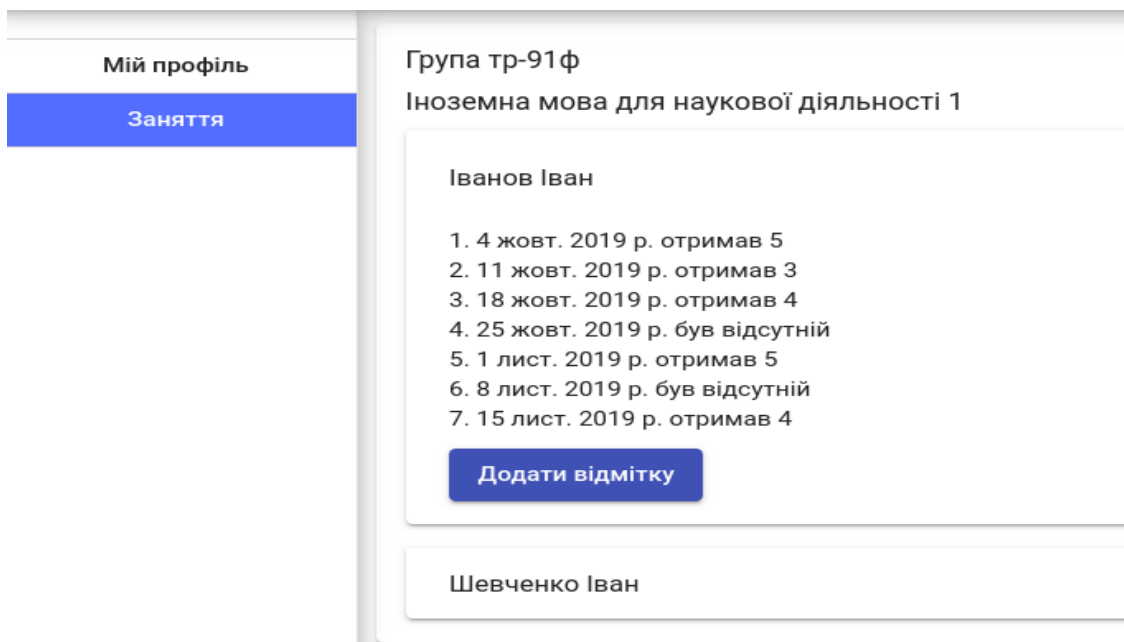
зберігаючи внесені у ньому зміни. Кнопка «Додати» закриває діалогове вікно та зберігає внесені зміни до інформаційної картки студента.



Група: тр-91ф
Предмет: Іноземна мова для наукової діяльності 1

Внести оцінки та відвідуваність

Рисунок 5.8 - Картка предмету



Мій профіль
Заняття

Група тр-91ф
Іноземна мова для наукової діяльності 1

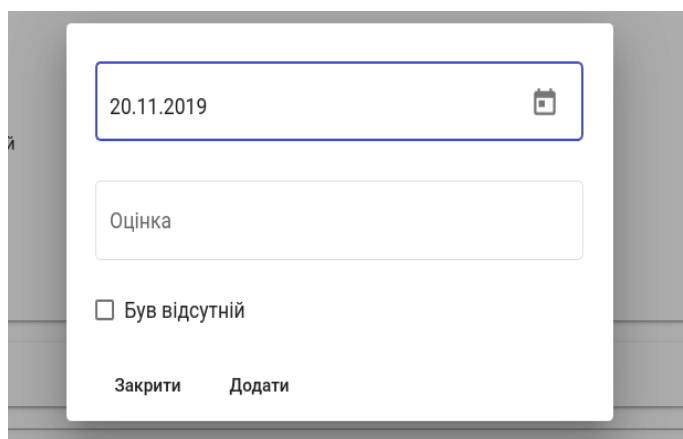
Іванов Іван

- 1. 4 жовт. 2019 р. отримав 5
- 2. 11 жовт. 2019 р. отримав 3
- 3. 18 жовт. 2019 р. отримав 4
- 4. 25 жовт. 2019 р. був відсутній
- 5. 1 лист. 2019 р. отримав 5
- 6. 8 лист. 2019 р. був відсутній
- 7. 15 лист. 2019 р. отримав 4

Додати відмітку

Шевченко Іван

Рисунок 5.9 - Розгорнута картка предмету



20.11.2019

Оцінка

☐ Був відсутній

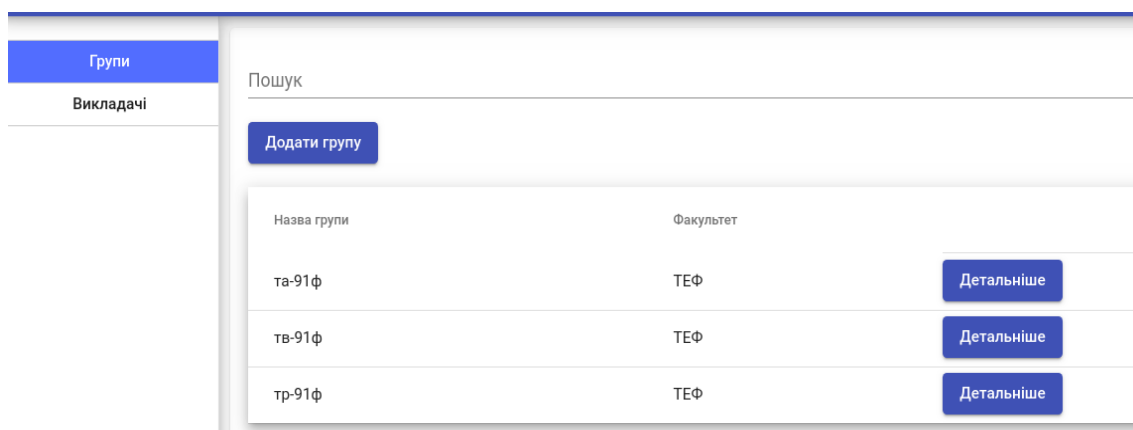
Закрити Додати

Рисунок 5.10 - Вікно додавання відмітки

5.4. Опис взаємодії з модулем адміністратора

Модуль адміністратора реалізовано у вигляді веб-додатку. Після успішної авторизації адміністратора перенаправляє до сторінки з навігаційним меню, оскільки модуль адміністратора не передбачає сторінку власного профілю. Навігаційне меню має лише два пункти, «Групи» та «Викладачі».

Пункт «Групи» дозволяє здійснювати адміністрування навчальних груп (рис.5.10). У верхній частині сторінки знаходиться поле пошуку групи, що дозволяє фільтрувати і відображати список груп. Якщо поле пусте, то у списку відображаються всі існуючі групи. Під полем пошуку знаходиться кнопка «Додати групу», при натисканні на яку, з'являється діалогове вікно з можливістю створення нової групи.



Пошук		
Додати групу		
Назва групи	Факультет	
та-91ф	ТЕФ	Детальніше
тв-91ф	ТЕФ	Детальніше
тр-91ф	ТЕФ	Детальніше

Рисунок 5.11 - Сторінка перегляду навчальних груп

Діалогове вікно містить поля вводу для назви групи, селектор вибору факультету та селектор вибору кафедри, а також кнопка «Додати групу» (рис.5.12).

Рисунок 5.12 - Вікно створення навчальної групи

Кожен елемент списку груп має в собі кнопку «Детальніше». При натисканні кнопки «Детальніше» адміністратора буде перенаправлено на сторінку редагування учбової діяльності групи (рис.5.13).

Рисунок 5.13 - Сторінка редагування учбової діяльності групи

Сторінка редагування учбової діяльності відображає список навчальних предметів групи посеместрово та має кнопку «Додати предмет». При натисканні на кнопку додати предмет відкривається діалогове вікно, що надає можливість

додати навчальний предмет до групи (рис 5.14). Діалогове вікно має наступні обов'язкові поля для заповнення:

- назва предмету;
- кількість кредитів;
- семестр навчання, обирається з випадаючого списку;
- викладач, що обирається з випадаючого списку

Назва предмету

Це поле обов'язкове

Кількість кредитів

Семестр
1

Викладач

- Коваль О.В.
- Шалденко О
- Препотенська М. П.
- Олізко Ю. М.

Рисунок 5.14 - Діалогове вікно додавання навчального предмету

У нижній частині діалогового вікна знаходяться дві кнопки, «Закрити» та «Додати». При натисканні на кнопку «Закрити», діалогове вікно закривається без збереження внесених в ньому даних. При натисканні кнопки «Додати» введені дані зберігаються, діалогове вікно закривається. Слід зауважити, що всі поля вводу у вікні мають додаткові поля-валідатори, що захищають від неправильного вводу даних або збереження даних з пустими значеннями.

У боковому меню під пунктом «Групи» знаходиться пункт «Викладачі».

Зверху сторінки знаходиться поле пошуку викладачів, під ним розташована кнопка «Додати викладача». Більшу частину інтерфейсу цієї сторінки займає

список існуючих викладачів, що представлений картками викладача (рис. 5.15).

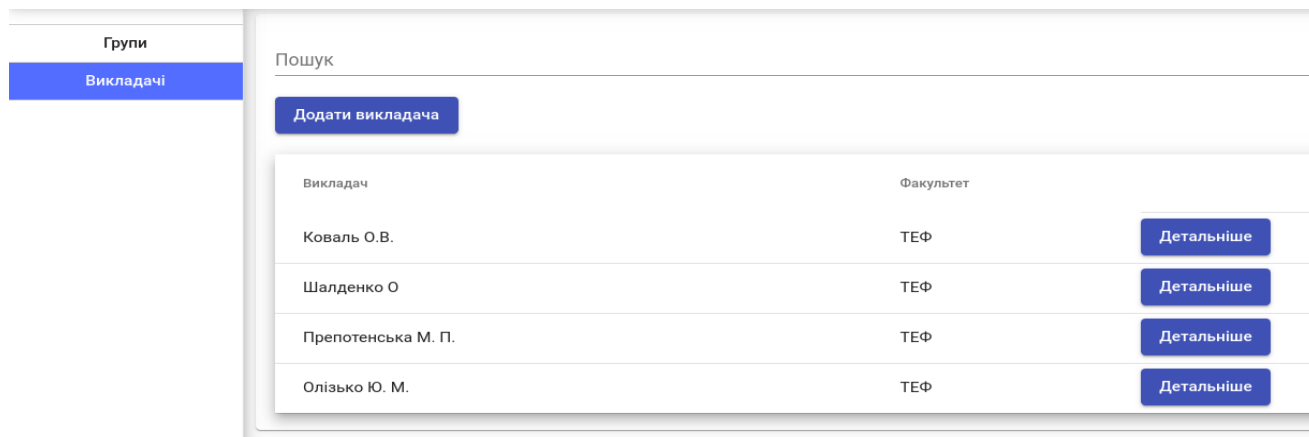


Рисунок 5.15 - Сторінка перегляду викладачів

Для того, аби скористатись полем пошуку викладачів, у нього потрібно внести прізвище та/або ініціали викладача, тоді список викладачів буде відфільтровано за введеними даними. Адміністратор має можливість реєструвати нових викладачів за необхідністю. Для цього адміністратор має натиснути кнопку «Додати викладача», тоді адміністратор автоматично буде перенаправлений до сторінки додавання нового викладача (рис.5.16).

Додати викладача

ПІБ викладача

email викладача
admin@admin.com

Пароль для викладача
.....

Факультет ▼

Кафедра ▼

Додати викладача

Рисунок 5.16 - Сторінка додавання(реєстрації) викладача

Функція додавання нового викладача має на увазі і автоматичну реєстрацію викладача у системі.

Для реєстрації викладача адміністратор має заповнити відповідну інформацію:

- ПІБ викладача, що реєструється;
- поштову адресу викладача;
- пароль для викладача;
- факультет реєстрації викладача, вибір факультету представлений випадającym списком;
- кафедру реєстрації викладача, вибір кафедри стає можливим лише після вибору факультету оскільки кафедри автоматично сортуються відповідно обраного факультету, представлений випадającym списком.

Після заповнення всіх інформаційних полів, внизу сторінки кнопка «Додати викладача» стане активною для взаємодії. Після її натискання викладач буде зареєстрований у системі, а адміністратор системи буде автоматично перенаправлений до сторінки «Викладачі».

Оскільки, при закінченні програмної сесії користувачем, система автоматично запам'ятовує останнього авторизованого користувача та при наступній роботі користувача з додатком, система виконає автоматичну авторизацію. Інструмент виходу, вилогіну, з системи знаходиться у верхньому правому кутку веб-сторінки, біля імені користувача, та загальний для будь-якого типу користувача (рис.5.17). При натисканні на кнопку виходу, користувач буде автоматично перенаправлений на сторінку авторизації, введення логіну та пароля.



Рисунок 5.17 - Панель виходу з облікового запису

5.5. Висновки до розділу

У розділі наведено опис системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації, що надає можливість адмініструвати навчальний процес вищого навчального закладу.

Перш за все адміністратор відповідає за реєстрацію нових користувачів та призначення їх ролі у системі. Адміністратор системи має можливість перегляду існуючих навчальних груп, редагування їх навчальної діяльності та створення нових. Редагування навчальної діяльності групи відбувається за рахунок додавання навчальних предметів до групи, та закріплення певного викладача за відповідним предметом.

Викладач має можливість з перегляду сторінки власного профілю, перегляду списку груп, де викладає, а також заповнення навчальної діяльності групи шляхом додавання навчальних відміток для кожного студента за відповідне заняття.

Студенту доступні можливості з перегляду власного профілю, індивідуального навчального плану, списку навчальних занять, розкладу пар, статистики успішності.

Загалом, під час тестування системи не було помічено логічно-наслідкових порушень системи, збоїв та інших несправностей, тому додаток можна використовувати на Android смартфонах та у веб-системах.

6. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

Розділ має на меті проведення маркетингового аналізу стартап проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження [21].

6.1 Опис ідеї системи

На сьогоднішній день дуже гостро стоїть питання всебічної автоматизації управління навчальним процесом. В умовах стрімкого розвитку інформаційних технологій потоки інформації, що циркулюють у світі, – величезні і мають тенденцію до збільшення. Тому, в будь-якій організації, виникає проблема керування даними, яке б забезпечило найбільш ефективну роботу. Не є виключенням і навчальні заклади.

Саме для навчального закладу - університету НТУУ «КПІ імені Ігоря Сікорського» (далі - в університеті) досі не існує комплексного рішення у вигляді системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації (далі - аспірантів). Це є великою проблемою, адже спеціалісти даного напрямку не мають доступу до важливої та необхідної для навчання інформації, зібраної в одному місці зі зручним інтерфейсом.

В університеті створена комплексна та інтегрована система управління, яка агрегує різні модулі та дані, що циркулюють між ними, пов'язані з навчанням та студентським життям. Однак, для аспірантів, на жаль, досі не існує зручного рішення.

На даний момент, користувачі цього напрямку мають досить обмежені можливості в рамках пошуку, перегляду, запиту, обміну, розміщення та аналізу необхідної інформації на базі існуючого веб-рішення, що надається університетом. Помітно, що данному ресурсу необхідні оновлення функціоналу.

Саме з розумінням даної проблеми і пропонується створення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації, в тому числі і розробка зручного та сучасного рішення у вигляді мобільного додатку на базі технології Android.

Ретельно проаналізувавши потреби спеціалістів науково-педагогічних кадрів вищої кваліфікації та університету, можна скласти наступну таблицю, яка характеризує зміст ідеї, напрямки застосування, а також вигоди для кожного типу користувача (табл.6.1).

Таблиця 6.1 Опис ідеї стартап-проекту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигоди для користувача</i>
Створення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації	1. Система як джерело актуальної інформації про навчання для аспірантів. 2. Система як канал зв'язку між аспірантами та викладачами, платформа для обміну інформацією. 3. Система як засіб донесення необхідних відомостей до користувачів, що ініціюється адміністрацією Відділу Аспірантури.	1. Система має бути зручною та зрозумілою у користуванні та повинна надавати аспірантам доступ до необхідних навчальних даних (розклад, успішність, профіль, індивідуальний план, важлива інформація, тощо). 2. Система має бути корисною для викладачів, надаючи необхідну та актуальну інформацію про розклад, графік викладання, успішність студентів, можливість

Продовження таблиці 6.1 Опис ідеї стартап-проекту

		<p>виставлення балів за успішність, тощо).</p> <p>3. Система повинна забезпечити можливість розміщення актуальної інформації для всіх користувачів ресурсу Адміністрацією. Це дозволить отримувати необхідні відомості всіма, хто цього потребує, в режимі онлайн.</p>
--	--	--

Наступним кроком було проведення аналізу наявних рішень в галузі, їх огляд та порівняння з запропонованим рішенням (табл.6.2).

Ринок українського програмного забезпечення пропонує наступні рішення управління навчальним процесом:

1. Автоматизована система керування вищим навчальним закладом АСУ «ВНЗ», розроблена Науково-дослідним інститутом Прикладних інформаційних технологій Кібернетичного центру Національної академії наук України.

2. Пакети комп'ютерних систем «Деканат», «БІБЛОГРАФ-2007», «КОЛОКВІУМ», «ПС-Персонал» приватного підприємства «Політек-софт».

3. «ВНЗ АЛЬМА-МАТЕР» товариства з обмеженою відповідальністю «Direct IT».

Таблиця 6.2 Визначення сильних, слабких та нейтральних характеристик

ідеї проекту

№ п/ п	Техніко- економічні характери- стики ідеї	(потенційні) товари/концепції конкурентів				W (слабк а сторо на)	N (нейтр альна сторон а)	S (сильна сторона)
		Мій проект	АСУ «ВНЗ»	«Політек -софт»	«ВНЗ АЛЬМА- МАТЕР»			
1.	Технологі- чність: 1.База даних 2.Мова програмув- ання 3.Викорис- тання бібліотек	1.Firebase cloud storage (хмара) 2.Kotlin, Angular, TypeScrip t 3.Сучасні: RxJava, RxJS, Dagger, Multibindi ng	1.SQL 2.Delphi 3.ADO, Midas, Internet	1.InterBas e 6.0, Firebird 2.Delphi 3. Data Access, Additiona l	1.MySQL 5.0 2.PHP 5.2 3.QReport, InternetBas e, Decision Cube	1.- 2.- 3.-	1.- 2.- 3.-	1.Використання хмарного рішення, що не потребує зовнішнього адмініструванн я, легка доступність, надійність, швидка інтеграція у проект. 2.Сучасна мова програмування, вища швидкість розробки. 3.Сучасні бібліотеки для dependency injection, data observing, user flow
2.	Автономні- сть	Не потребує зовнішнь ої підтримк и з боку розробни- ка (backend)	Потребу- є окремо виділено- го розробни- ка (backend)	Звертаєть- ся до послуг системн ого адміністр атора кожного разу, коли необхідн о налаштув ати мережу	Потребує окремо виділеного розробник а (backend)	-	-	Система може підтримуватися основними розробниками продукту без додаткового залучення інших спеціалістів

Продовження таблиці 6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

3.	Крос-платформеність	Має як веб-інтерфейс, так і Андроїд-додаток	Складається з підсистем (пакетів) лише у «коробковому» форматі.	Складається з підсистем (пакетів), але є програмні засоби, які дозволяють здійснювати динамічну генерацію Web-сторінок для мережі Internet на основі даних, що фіксуються Пакетами	Програмний комплекс, що складається з модулів.	-	Сама система являє собою самостійний модуль-рішення для роботи саме Аспірантського відділу.	Зручність у використанні веб-додатку, що не висуває суворих вимог до апаратного забезпечення. Додаткова цінність для користувачів у вигляді окремого Андроїд-додатку.
4.	Гнучкість	Програмні модулі можна легко редагувати, не руйнуючи поточну роботу системи	Не передбачається технічного втручання та оновлення після купівлі системи.	Є можливість підключення додаткових пакетів-банків даних для введення та збереження інформації	Система може бути пристосована до структури різних навчальних закладів.	-	Необхідні кваліфіковані розробники та сильна експертиза для створення та підтримки гнучкості.	Перевага у відсутності необхідності залучення додаткових спеціалістів. Можливість редагування та модифікації без необхідності перебудови всієї системи.

Продовження таблиці 6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

5.	Економічність	Не потребує придбання власного сервера для розташування бази даних в мережі Інтернет. Витрати - на заробітну платню розробникам, які зроблять продукт у відповідності з технічним завданням.	В освітніх закладах система впроваджувалася безкоштовно до 2006 року (за умовами партнерства).	Різні функціональні модулі купуються окремо і мають різні ціни, що відповідають обраному функціоналу.	Система складається з декількох модулів, кожен з яких можна придбати окремо або всю систему в цілому. Модулі можуть бути кастомізовані під потреби замовника.	Може коштувати порівняно дорожче, аніж існуючі готові універсальні рішення.	Окрім витрат на розробку, інших витрат не передбачається (сервер, база даних, додаткові спеціалісти)	Витрати на розробку системи великою ймовірністю будуть виправдані зручністю та кастомізованістю функціоналу системи.
6.	Адміністрування	У Адміністратора за замовчуванням є весь необхідний спектр повноважень в рамках модерації контенту, контролю та адміністрування системи в	Необхідно навчитися адмініструвати систему, дотримуючись правил даного типового рішення. В функціоналі закладено лише	Система може бути адміністрована навченими спеціалістами. Використовується початкові набори команд та можливо стей адміністратора без	Система є досить сучасним рішенням, тому, набір функціоналу може вважатися достатнім, а навчання особливостям системи не повинне бути дуже складним.	-	Достатня кількість можливих дій адміністратора з системою.	Використання сучасних технологій розробки спрощують задачу адміністрування та модифікації- за вимогами адміністратора система може бути налагоджена більш зручним чином. Важливо також, що навчити

Продовження таблиці 6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

		цілому. Легке навчання, що є важливим аспектом для працівників з різними навичками користування ПК.	обмежений набір команд, що можуть виконуватися адміністратором. В контексті сучасних вимог до ПЗ в університеті, такого набору повноважень може бути недостатньо.	можливо сті гнучкого конфігурування.				користуватися системою має
7.	Модульність	Система технічно представляє собою автономний модуль для керування навчальним процесом аспірантів. Розрахована на три типи користувачів: Адміністр	Підсистема «Приймальна комісія», Підсистема «Деканат» складають систему. Дані підсистеми не вирішують досліджу	Система складається з наступних підсистем: Підсистема «ПС-Персонал», Підсистема «БІБЛЮГРАФ-2007», Підсистема	Інформаційне середовище, яке базується на сучасних web-технологіях має наступні модулі у своєму складі: «Приймальна комісія», «Деканат		Система в цілому є автономним модулем, а не комплексним рішенням для всього керування інформацією, що циркулює	Система легко може бути інтегрована до загальноуніверситетської АСУ. Система задовольняє потреби цільових користувачів (Адміністратор, Аспірант, Викладач) та вирішує визначену проблему.

Продовження таблиці 6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

		атор, Аспірант, Викладач. В той самий час, може бути приєднан а разом з даними до глобально ї АСУ університ ету.	вану проблем у.	«Деканат », Підсисте ма «Колоквіу м». Жодна з них не вирішує проблему супровод ження навчання аспірантів .	навчальна частина», «Учбово- методичн ий відділ». Кожна частина може лише частково задовольн ити інформаці йні потреби Аспірантс ького Відділу університ ету.		ює в універс итеті.	
--	--	---	-----------------------	--	---	--	---------------------------	--

Провівши даний аналіз слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару, - в умовах поставленого завдання, дану систему можна вважати конкурентоздатною у порівнянні з існуючими рішеннями в галузі.

6.2 Технологічний аудит ідеї проекту

Для визначення технологічної здійсненості ідеї проекту було проаналізовано наступні параметри (табл.6.3):

- можливі технології, котрі будуть використані в процесі реалізації

системи;

- наявність технологій/необхідність розробки;
- доступність технологій.

Таблиця 6.3. Технологічна здійсненність ідеї проекту

<i>№ n/n</i>	<i>Ідея проекту</i>	<i>Технології її реалізації</i>	<i>Наявність технологій</i>	<i>Доступність технологій</i>
1.	Аутентифікація та автоматична реєстрація користувача за допомогою електронної пошти	Google Cloud Messages	Наявна технологія	Доступна
2.	Створення та редагування навчальних груп адміністратором системи	Angular, Firebase Firestore	Наявна технологія	Доступна
3.	Створення крос-платформеного рішення для типу користувача: Аспірант.	Angular, Android, Firebase Firestore, Kotlin	Наявна технологія	Доступна
4.	Система внесення відміток про успішність Аспіранта	Angular, TypeScript, Firebase Firestore	Наявна технологія	Доступна
5.	Інтеграція системи розкладу для навчальних груп	«Розклад КПІ», Angular, TypeScript, Android, Kotlin	Наявна технологія	Доступна
6.	Створення медіа-бібліотеки для збереження книжок	Angular, TypeScript, DatabaseRealtime, Firebase Storage	Наявна технологія	Загально-недоступна
7.	Авторизація за допомогою соціальних мереж	GoogleAuth, TwitterOAuth, FacebookOAuth	Наявна технологія	Загально-недоступна

Таким чином, з технологічної точки зору реалізація системи є можливою з включенням (але не обмеженням) основного функціоналу під номерами 1-5 в таблиці 6.3.

Ідеї під номерами 6-7 визначені як бажані, але не обов'язкові. На їхню реалізацію може бути виділено набагато більше ресурсів, ніж буде отримано користі, що вважається неефективним.

Отже, нижче узагальнено обрані технології реалізації ідей проекту:

- Середовище розробки: Android Studio, Visual Studio;
- Мови програмування (об'єктно-орієнтована): TypeScript, Kotlin;
- Frameworks: Android, Angular;
- Бібліотеки: RxJava, Dagger, Firestore, Location API, Google Services;
- База даних: NoSQL Database (Firebase Firestore).

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Наступним кроком є аналіз та опис потенційного ринку стартап-проекту (таблиця 6.4): наявність попиту, обсяг, динаміка розвитку ринку.

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	1 головний гравець
2.	Загальний обсяг продаж, грн/ум.од	\$1 млрд
3.	Динаміка ринку (якісна оцінка)	низька
4.	Наявність обмежень для входу (вказати характер обмежень)	затвердження ліцензійних умов провадження, створення маркетингової стратегії для проведення ефективної рекламної діяльності стосовно ПЗ

Продовження таблиці 6.4. Попередня характеристика потенційного ринку стартап-проекту

5.	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	+15%

На основі даних, отриманих в результаті аналізу потенційного ринку, можна прийти до висновку, що так як середня норма рентабельності по ринку (15%) у порівнянні з банківським відсотком на вкладення (5%) є сенс вкладати гроші в даний проект. Отже, ринок є привабливим для входження за попереднім оцінюванням.

Надалі необхідно визначити потенційні групи клієнтів, їх характеристики, та сформувані орієнтовний перелік вимог до товару для кожної групи (табл. 6.5).

Таблиця 6.5 Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Створення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних	Відділ Аспірантури «КПІ», спеціалісти науково-педагогічних кадрів вищої кваліфікації, викладачі університету	Відділ Аспірантури «КПІ» потребує сучасної автоматизованої системи управління навчальним процесом, у	Компонент повинен забезпечити зручний програмний інтерфейс для керування навчальним процесом спеціалістів

Продовження таблиці 6.5 Характеристика потенційних клієнтів стартап-проекту

	кадрів вищої кваліфікації		той час як науково-педагогічні кадри, що здійснюють свою підготовку потребують зручної системи для відстеження учбового процесу.	науково-педагогічних кадрів. Забезпечити додаткову реалізацію програмного рішення задля здобуття крос-платформних можливостей.
--	---------------------------	--	--	--

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: було складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 6.6-7).

Таблиця 6.6 Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Вибір іншого програмного забезпечення, що надає схожі можливості	Керівництво вищого навчального закладу може надати перевагу вже готовій реалізації від іншого виробника	Продовжувати розробку програмного комплексу, оскільки дане рішення можливо інтегрувати не лише в одному навчальному закладі.
2.	Планування та послідовність виконання задач	Навіть якщо ці завдання виконують різні люди, їх одночасне виконання у великій кількості створює ризик для проекту, особливо наприкінці його реалізації.	Перевірити, чи не заплановано забагато завдань на один і той самий час. При плануванні задач проекту спочатку необхідно скласти список завдань і згрупувати їх, щоб оцінити весь обсяг проекту та кінцеві результати. Потім можна починати зв'язувати завдання, щоб отримати ідеальний розклад.

Таблиця 6.7 Фактори можливостей

№ п/п	Фактор	Зміст	Можлива реакція компанії
1.	Аналіз досвіду конкурентів	Формування стратегії реалізації проекту без навчання на своїх помилках, а при навчанні на помилках конкурентів – невдалі рекламні, маркетингові ходи конкурентів.	Планування і реалізація проекту з максимальним виключенням ймовірності виникнення помилок вже досвідчених конкурентів на ринку споживачів.
2.	Націлення продукту на основні функціональності, які відсутні у конкурентів	Реалізація нових можливостей для споживачів, впровадження покращень суміжного з конкурентами проекту функціональностями.	Чітке планування задач, розподілення задач між розробниками з залученням розрахованих ризиків, мотивація команди ідеєю кінцевого продукту.
3.	Підвищення рентабельності проекту	За рахунок правильного планування всіх етапів проекту, чіткого формулювання бізнес-моделі є можливість залучення до команди проекту студентів в якості розробників.	Зниження кількості інвестицій для розробки і впровадження кінцевого продукту.

Далі було проведено аналіз пропозиції: визначено загальні риси конкуренції на ринку (табл. 6.8).

Таблиця 6.8 Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>
1. Олігополістична конкуренція	Галузь в основному є конкурентною, проте існує декілька явних лідерів	Важко вийти на міжнародний рівень
2. Локальний рівень конкурентної боротьби	Конкуренти з України	Розвиток на українській ІТ арені та вихід на ринок
3. Внутрішньогалузева конкуренція	Конкуренція спостерігається в пропозиціях на покупку програмного забезпечення (вигідні пропозиції), якості функціональностей.	Розробка вузьконаправленого програмного забезпечення
4. Конкуренція за видами товарів: товарно-видова	Конкуренція між програмними забезпеченнями одного виду	Випуск кращих і якісніших версій програмного забезпечення, взаємодія з пропозиціями і побажаннями споживача.
5. За характером конкурентних переваг – нецінова конкуренція	Функціональні можливості програмного забезпечення	Розширити функціональні можливості
6. За інтенсивністю – не марочна конкуренція	Для споживачів має значення якість виконаної розробки	Створення надійного та функціонального програмного забезпечення

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера) (табл. 6.9).

Таблиця 6.9 Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	1. АСУ» ВНЗ» 2. ПП «Політек - СОФТ» 3. Програмний комплекс «АЛЬМА - МАТЕР»	- Можливість здійснення вибору на користь існуючого рішення в галузі - Недостатнє фінансування, нечітке ТЗ.	Основним постачальником можна визначити інтернет-ресурси та література з програмування, які не суттєво впливають на проект.	Відділ Аспірантури «КП», спеціалісти науково-педагогічних кадрів вищої кваліфікації, викладачі університету	На ринку наявні лише готові універсальні рішення, які для наших цілей мають бути додатково дороблені та кастомізовані. Рішення, яке б на 100% задовольняло потреби саме Аспірантського відділу, - відсутнє.
Висновки :	Конкурентна боротьба неінтенсивна так, як прямі конкуренти більше спеціалізуються на інших функціональних можливостях, готових рішеннях.	Є можливості входу на ринок за рахунок більш ймовірного досягнення цілей; потенціал конкуренції є серед існуючих компаній	Зазвичай постачальники не диктують умови співпраці, так як для реалізації проекту не потрібне постачання матеріалів, потребується лише експертиза спеціалістів.	Рішення створюється конкретно для відділу Аспірантури університету, може застосовуватися не тільки в КП. Однак, так як проект є досить специфічним, влада Клієнтів є значною.	Відсутні. Є лише універсальні рішення в галузі.

Таким чином, завдяки попередньому аналізу, а також з огляду на конкурентну ситуацію можна визначити, що є велика ймовірність успішного

входження до ринку. Крім того, сильні сторони системи, що були визначені в ході аналізу, можна вважати конкурентоспроможним на ринку.

На основі аналізу конкуренції, із урахуванням характеристик ідеї проекту, вимог споживачів до товару та факторів маркетингового середовища можна визначити та обґрунтувати перелік факторів конкурентоспроможності (табл. 6.10).

Таблиця 6.10 Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Потреби споживачів	Потреби споживачів обумовлюють необхідність розробки проекту. Якісний аналіз таких потреб сприяє створення точного і цілісного технічного завдання.
2.	Результативність	Завжди досягається кінцевий результат, кожен проміжний результат є автономною та робочою версією продукту.
3.	Ціна та собівартість продукції	Нижча ринкова ціна досягається використанням сучасних технологій розробки.
4.	Технічне обслуговування	Випуск нових версій продукту, оновлення технічної бази.

За визначеними факторами конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 6.11).

Таблиця 6.11 Порівняльний аналіз сильних та слабких сторін програмного забезпечення системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації

№ п / п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг програм-конкурентів у порівнянні з системою супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації						
			-3	-2	-1	0	+1	+2	+3
1.	Потреби споживачів	5		+					
2.	Результативність	12				+			
3.	Ціна та собівартість продукції	10			+				
4.	Технічне обслуговування	5		+					

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 6.12).

Таблиця 6.12 SWOT- аналіз стартап-проекту

Сильні сторони: Простота та зручність у використанні, кросплатформеність, привабливий інтерфейс користувача.	Слабкі сторони: Відсутність досвіду використання програмного продукту, можлива присутність багів.
Можливості: Вихід на національний ринок; розширення функціональних можливостей та доопрацювання поточних.	Загрози: Присутність недоробок на ранньому етапі може спричинити відтік клієнтів.

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 6.13).

Таблиця 6.13 Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Створення системи як автономного модулю, призначеного для функціонування лише в університеті КПП.	75%	9 міс.
2.	Створення системи як універсального модулю, призначеного для впровадження не тільки в університеті КПП.	50%	12 міс.
3.	Створення системи на основі існуючого рішення із модифікацією деяких властивостей.	30%	6 міс.

Таким чином, проаналізувавши три вищенаведені альтернативи було обрано альтернативу №1, так як вона є кращою та найоптимальнішою за наступними показниками:

- а) отримання ресурсів є більш простим та ймовірним;
- б) строки реалізації – більш стислими.

6.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії перш за все передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 6.14).

Таблиця 6.14 Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту) за рік	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Вищі навчальні заклади	Готові	середній	Невелика конкуренція	Легка
2.	Університет КПІ	Готові	високий	Мала конкуренція	Легка

Які цільові групи обрано: обрано цільову групу - університет КПІ, так як саме для даної групи є можливість задоволення 100% інтересів клієнтів завдяки створенню системи супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації. У цієї групи також є високий попит на дану систему, та відносно мала конкуренція, звідки простота входу у сегмент - легка.

За результатами аналізу потенційних груп споживачів необхідно обрати цільові групи, для яких буде запропоновано продукт, також необхідно визначити стратегію охоплення ринку, базову стратегію розвитку (табл.6.15).

Таблиця 6.15 Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Створення системи як автономного модулю, призначеного для функціонування лише в університеті КПІ.	Стратегія концентрованого маркетингу	Низька ціна, легке встановлення системи, максимальна відповідність потребам користувачів системи.	Стратегія спеціалізації

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 6.16).

Таблиця 6.16 Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Проект не є першопрохідцем, досі існує декілька рішень, які є універсальними для навчальних закладів	Компанія буде забирати існуючих споживачів у конкурентів і шукати нових	Основні характеристики товару будуть схожими до характеристик конкурентів, але більш специфічними	Стратегія заняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробляється стратегія позиціонування (табл. 6.17). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати проект.

Таблиця 6.17 Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Зручність у використанні, низька вартість	Знизити ціни на продукцію та створити систему, що буде зручна у використанні, кросплатформена	Зручність інтерфейсу та легкість використання, кросплатформеність, низька ціна	Сучасність, швидкість, стабільність, надійність, зручність.

Отже, завдяки дослідженням, проведеним в даному підрозділі, було створено узгоджену систему рішень щодо ринкової поведінки стартап-компанії, яка визначатиме напрями роботи стартап-компанії на ринку. А саме:

- Цільова група: університет КПП;
- Альтернатива розвитку проекту: створення системи як автономного модулю, призначеного для функціонування лише в університеті КПП;
- Стратегія охоплення ринку: стратегія концентрованого маркетингу;
- Базова стратегія розвитку: стратегія спеціалізації;
- Стратегія конкурентної поведінки: стратегія заняття конкурентної ніші.

6.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 6.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 6.18 Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує ПЗ	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Легкість у використанні, інтуїтивність	Легкий інтерфейс для розуміння процесу роботи системи	Створення сучасних інтерфейсів відповідно за допомогою найкращих довідників з побудови інтерфейсу користувача
2.	Доступність	Кросплатформне рішення, оскільки як веб-додаток, так і мобільний додаток	Доступність до системи у будь-який момент
3.	Адміністрування	Для встановлення системи не потрібно великих матеріально-технічних витрат на встановлення серверів	Легкість розгортання системи

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту, його фізичні складові, особливості процесу його надання (табл. 19).

Таблиця 6.19 Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за заду- мом	Створити систему супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Надання чіткого зрозумілого інтерфейсу для керування, кросплатформеність. 2. Реєстрація та авторизація користувачів у системі. 3. Можливість створення та перегляду навчального плану. 4. Можливість перегляду навчального розкладу. 5. Можливість створення навчальних груп з призначенням навчальних предметів та відповідних викладачів. 6. Можливість ведення успішності для кожного студента. 7. Можливість перегляду статистики власної успішності.	-	-
	Якість: стандарти, нормативи, параметри тестування		
	Марка: «Аспірантура КПІ»		
III. Товар із підкріпленням	До продажу: презентація програмного продукту Після продажу: підтримка клієнтів, розширення існуючого функціоналу		
За рахунок чого потенційний товар буде захищено від копіювання: закритий код та ліцензування			

Наступним кроком є визначення оптимальної системи збуту, в межах якої приймається рішення (табл. 6.20):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту; вибір та обґрунтування виду посередників.

Таблиця 6.20 Формування системи збуту

№ п/п	Специфіка поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Клієнти купують програмне забезпечення безпосередньо у компанії-розробника	Налаштування програмного забезпечення на матеріально-технічній базі замовника	Канал нульового рівня (виробник безпосередньо продає товар клієнту)	Через адміністрацію клієнта

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 6.21).

Таблиця 6.21 Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Клієнти дізнаються про нові модулі системи з допомогою поштових розсилок	Інтернет, соціальні мережі	Низька ціна, зручність у використанні	Поширення знань про нові програмні модулі/оновлення.	<ul style="list-style-type: none"> - Перелік основних правдивих даних про оновлення програмного забезпечення, створення нових модулів системи - Науково-професійний стиль

Як результат, було визначено ринкову (маркетингову) програму, що включає в себе концепції товару, збуту, просування, що спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого буде впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

6.6 Висновки до розділу

В результаті створення маркетингового аналізу стартап-проекту були визначені загальні напрями використання потенційного програмного продукту, а також відмінність від конкурентів.

Було визначено такі показники стартап-проекту:

- на даний програмний програмний продукт існує попит, це означає що проект можна комерціалізувати;
- реалізація продукту та його імплементація є доцільною, оскільки передбачає подальше накопичення додаткового функціоналу;
- ринок конкуренції низький, а попит на систему- високий, перспектива впровадження продукту у разі успішної реалізації - висока.

ВИСНОВКИ

В результаті дослідження була створена система супроводження та аналізу навчального процесу науково-педагогічних кадрів вищої кваліфікації.

Перед створенням системи було проаналізовано існуючі рішення, що вирішували задачу автоматизації навчального процесу. До уваги було прийнято функціональний набір програмних продуктів і їхні структури, позитивні та негативні характеристики систем. Незважаючи на велику кількість варіантів, доступних для використання, було зроблено висновок про недостатність існуючих рішень у межах поставленого завдання.

Для реалізації системи супроводження та аналізу навчального процесу науково-педагогічних кадрів вищої кваліфікації були обрані передові технології реалізації, завдяки ним створена система є надійною та легкою у підтримці, передбачає легке розширення майбутнього функціоналу та масштабування системи загалом.

Розроблена система у повному обсязі покриває поставлені завдання, оскільки перед її розробкою було виокремлено основні критерії, яким мають відповідати програмні модулі системи. Внаслідок чого система вирішує проблеми трьох основних типів користувачів. Для кожного типу користувача було поставлено функціональні вимоги, що є актуальними та нагальними, і були визначені завдяки взаємодії з відділом Аспірантури КПІ.

Розроблена система є кросплатформною, саме тому вона може виконуватись на великій кількості платформ. Запуск системи передбачає наявність у користувача сучасного браузеру або мобільної операційної системи Android.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Концепція інформатизації діяльності університету [Електронний ресурс]. — Режим доступу: https://kneu.edu.ua/userfiles/norm_doc/concept_it2013.pdf
2. Навчання в аспірантурі [Електронний ресурс]. — Режим доступу: <http://aspirantura.kpi.ua/>
3. ПП «Політек-софт» [Електронний ресурс]. — Режим доступу: <http://www.politek-soft.kiev.ua>
4. Білощицький А.О. Методи та моделі комплексного інформаційно-освітнього середовища в умовах розвитку вищого навчального закладу: Автореф... дис. канд. техн. наук: 05.13.06 — Київ, 2007. — 23с.
5. Науково-дослідний інститут прикладних інформаційних технологій [Електронний ресурс]. — Режим доступу: <http://www.ndipit.com.ua>
6. Програма автоматизації ВНЗ «АЛЬМА-МАТЕР» [Електронний ресурс]. — Режим доступу: <http://www.alma-mater.org.ua>
7. Можливості Android Studio [Електронний ресурс]. — Режим доступу: <https://developer.android.com/studio/intro>
8. Б. Харди, Б. Филлипс, К. Стюарт, К. Марсикано — «Андроид для профессионалов», 2014 р. — 475 с.
9. Android World [Електронний ресурс]. — Режим доступу: <https://androidworld.com>
10. Портнякин И. — «Kotlin in Action», 2015 р. — 300 с.
11. Руководство по языку Kotlin [Електронний ресурс]. — Режим доступу до ресурсу: <https://metanit.com/kotlin/>
12. Kotlin onboarding [Електронний ресурс]. — Режим доступу до ресурсу: <https://kotlinlang.org>

13. Основні принципи ін'єкції залежностей [Електронний ресурс]. — Режим доступу: <https://www.tutorialsteacher.com/ioc/dependency-injection>
14. Швидкий вступ до ін'єкції залежностей [Електронний ресурс]. — Режим доступу: <https://www.freecodecamp.org/news/a-quick-intro-to-dependency-injection-what-it-is-and-when-to-use-it-7578c84fa88f/>
15. Dagger for Android [Електронний ресурс]. — Режим доступу: <https://dagger.dev/android.html>
16. Using Dagger in your Apps [Електронний ресурс]. — Режим доступу: <https://developer.android.com/training/dependency-injection/dagger-android>
17. Бен Кристенсен — «Reactive Programming with RxJava: Creating Asynchronous», 2014 р. — 560 с.
18. Відкриваємо RxJava [Електронний ресурс]. — Режим доступу: <https://proandroiddev.com/exploring-rxjava-in-android-e52ed7ef32e2>
19. Документація по Firestore [Електронний ресурс]. — Режим доступу: <https://metanit.com/firestore/>
20. Документація по Firestore [Електронний ресурс]. — Режим доступу: : <https://www.firestore.org/docs/home.html>
21. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. — Київ : НТУУ «КПІ», 2016. — 28 с.

Додаток А
Апробація наукових досліджень

Система супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації

Апробації

УКР.НТУУ «КПІ ім. Ігоря Сікорського».ТМ-4227

Аркушів 5

2019

www.konferenciaonline.org.ua

**Міжнародна наукова
інтернет-конференція**

**Інформаційне суспільство:
технологічні, економічні
та технічні аспекти становлення**

(випуск 44)

ISSN 2522-932X

12 грудня 2019 р.

Тернопіль
2019

Зміст

Секція 1. Інформаційні системи і технології

Ваш Ю.В.

Дизайн та технології створення ігрового рушія..... 3

Верещак О.Р.

Прогнозування параметрів комп'ютерної мережі на основі
інтелектуальних технологій5

Врадій Д.В.

Система супроводження та аналізу вступу спеціалістів науково-
педагогічних кадрів вищої кваліфікації..... .8

Гумінський А.М.

Система супроводження та аналізу навчального процесу спеціалістів
науково-педагогічних кадрів вищої кваліфікації..... 9

Гоголь В.М., Гуменюк Т.В.

Комп'ютерна система ідентифікації технологічних станів бурової
установки 12

Дмитрієв В.С.

Комп'ютер та комп'ютерні системи..... 14

Мінько А.Ю.

Інформаційна безпека і засоби захисту інформації..... 16

Стелюк Б.Б., Олійник О.О.

Підходи до оцінки загроз інформаційної безпеки організації..... 17

Гумінський А.М., магістрант
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», м. Київ.
Кафедра автоматизації проектування енергетичних процесів і систем

СИСТЕМА СУПРОВОДЖЕННЯ ТА АНАЛІЗУ НАВЧАЛЬНОГО ПРОЦЕСУ СПЕЦІАЛІСТІВ НАУКОВО-ПЕДАГОГІЧНИХ КАДРІВ ВИЩОЇ КВАЛІФІКАЦІЇ

Існуючі на ринку програмні продукти, як правило, слабо враховують яскраво виражену галузеву специфіку ринку освіти. До того ж через брак ресурсів проекти комплексної автоматизації в ВНЗ часто гальмуються і не доводяться до кінця. Деякі ВНЗ йдуть шляхом створення власних розробок силами фахівців ІТ-підрозділів, що дозволяє створювати рішення, що враховують особливості бізнес-процесів конкретного закладу. Але плюси такого підходу дуже часто повністю нівелюються великою залежністю від команди програмістів. ВНЗ сьогодні працюють в умовах жорсткої конкуренції, більш того - змушені щодня підтверджувати свою затребуваність на ринку освітніх послуг і фінансову спроможність. Саме тому виникає потреба у створенні системи автоматизації управління навчальним процесом спеціалістів науково-педагогічних кадрів вищої кваліфікації, адже сьогодні, за стрімкого розвитку інформаційних технологій, потоки інформації, якими оперує людина, збільшуються щодня.

У зв'язку проблемою налагодження ефективного управління процесом навчання було вирішено розробити три функціональні підсистеми:

- підсистема адміністратора;
- підсистема аспіранта;
- підсистема викладача.

У системі користувач переважно займає позицію спостерігача та має можливість лише слідкувати за перебігом події, у той час коли адміністратор системи та викладачі мають право на внесення змін до інформаційної бази

шляхом взаємодії з веб-додатком. На рисунку 1 зображено функціональні можливості системи.

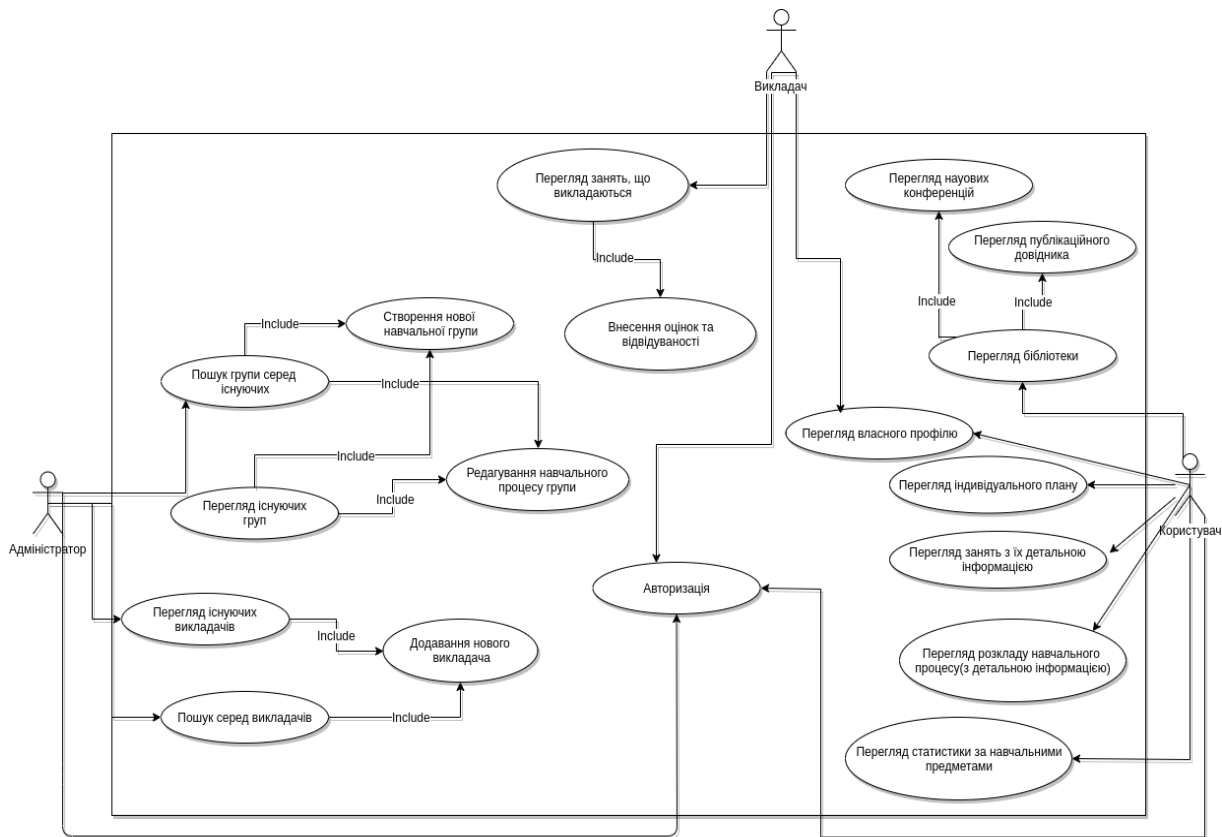


Рис. 1. Діаграма прецедентів системи

Одним з головних виборів при розробці програмного продукту є вибір програмних технологій реалізації продукту. Для реалізації Андроїд додатку було використано мову програмування Kotlin [1], для реалізації веб-додатку було використано мову TypeScript [2]. Створені веб-додаток та мобільний додаток містять сервіси та компоненти для роботи з віддаленою базою даних Firestore [3], реалізують асинхронну роботу за допомогою сучасної бібліотеки Rx [4] та реалізують такий шаблон програмування, як Dependency Injection [5].

Обрані технології реалізації є передовими та користуються великим попитом на ринку праці, програмні комплекси, що є продуктом їх використання є надійними, легко підтримуються та оновлюються.

Література

1. Документація до мови Kotlin [Електронний ресурс]. — Режим доступу до ресурсу: <https://metanit.com/kotlin/tutorial/>
2. Документація до мови TypeScript [Електронний ресурс]. — Режим доступу до ресурсу: [https://metanit.com /web/typeScript/](https://metanit.com/web/typeScript/)
3. Документація по Firestore [Електронний ресурс]. — Режим доступу: <https://metanit.com/firestore/>
4. Відкриваємо RxJava [Електронний ресурс]. — Режим доступу: <https://proandroiddev.com/exploring-rxjava-in-android-e52ed7ef32e2>
5. Основні принципи ін'єкції залежностей [Електронний ресурс]. — Режим доступу: <https://www.tutorialsteacher.com/ioc/dependency-injection>

Тема: Система супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації

Виконала: магістрант групи ТМ-81мп Гумінський Артем Миколайович

Науковий керівник: к. т. н., доцент Коваль Олександр Васильович

Ключові слова: *НАВЧАННЯ, АСПІРАНТ, ВИКЛАДАЧ, АДМІНІСТРАЦІЯ, КРОСПЛАТФОРМНІСТЬ, МОБІЛЬНИЙ ДОДАТОК, ANDROID ДОДАТОК, ВЕБ-ДОДАТОК, ХМАРНА ТЕХНОЛОГІЯ.*

Анотація на 3-х мовах.

Існуючі на ринку програмні продукти, як правило, слабо враховують яскраво виражену галузеву специфіку ринку освіти. До того ж через брак ресурсів проекти комплексної автоматизації в ВНЗ часто гальмуються і не доводяться до кінця. Деякі ВНЗ йдуть шляхом створення власних розробок силами фахівців ІТ-підрозділів, що дозволяє створювати рішення, що враховують особливості бізнес-процесів конкретного закладу. Але плюси такого підходу дуже часто повністю нівелюються великою залежністю від команди програмістів. ВНЗ сьогодні працюють в умовах жорсткої конкуренції, більш того - змушені щодня підтверджувати свою затребуваність на ринку освітніх послуг і фінансову спроможність. Саме тому виникає потреба у створенні системи автоматизації управління навчальним процесом спеціалістів науково-педагогічних кадрів вищої кваліфікації, адже сьогодні, за стрімкого розвитку інформаційних технологій, потоки інформації, якими оперує людина, збільшуються щодня.

Существующие на рынке программные продукты, как правило, слабо учитывают ярко выраженную отраслевую специфику рынка образования. К тому же из-за нехватки ресурсов проекты комплексной автоматизации в вузах часто тормозятся и не доводятся до конца. Некоторые вузы идут по пути создания собственных разработок силами специалистов IT-подразделений, позволяет создавать решения, учитывающие особенности бизнес-процессов конкретного заведения. Но плюсы такого подхода очень часто полностью нивелируются большой зависимостью от команды программистов. Вузов сегодня работают в условиях жесткой конкуренции, более того - вынуждены ежедневно подтверждать свою востребованность на рынке образовательных услуг и финансовую состоятельность. Именно поэтому возникает потребность в создании системы автоматизации управления учебным процессом специалистов научно-педагогических кадров высшей квалификации, ведь сегодня, по стремительного развития информационных технологий, потоки информации, которыми оперирует человек, увеличиваются ежедневно.

Existing software products, as a rule, have little regard for the pronounced industry specificity of the education market. In addition, due to lack of resources, complex automation projects in universities are often slowed down and not completed. Some universities go through the creation of their own developments by specialists of IT-departments, which allows to create solutions that take into account the peculiarities of business processes of a particular institution. But the benefits of this approach are very often completely offset by a great dependence on the programming team. Universities today operate in conditions of fierce competition, moreover - they are forced to confirm their demand on the market of educational services and financial capacity on a daily basis. That is why there is a need to create a system of automation of management of the educational process of specialists of scientific and pedagogical staff of higher qualification, because today, with the rapid development of information technologies, the flows of information operated by people are increasing daily.

Короткий опис.

В результаті була розроблена система супроводження та аналізу навчального процесу науково-педагогічних кадрів вищої кваліфікації.